TEMPERATURE HISTORIES IN THIN STEEL
PLATE DURING GAS METAL ARC WELDING

Ralph Hall Lipfert

TEMPERATURE HISTORIES IN THIN STEEL PLATE DURING GAS METAL ARC WELDING

by

RALPH HALL LIPFERT

B.S., United States Naval Academy

(1966)

SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

OCEAN ENGINEER

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 1972

TEMPERATURE HISTORIES IN THIN STEEL PLATE

DURING GAS METAL ARC WELDING

by

RALPH HALL LIPFERT

Submitted to the Department of Ocean Engineering in partial fulfillment of the requirements for the degree of Ocean Engineer.

ABSTRACT

During gas metal arc welding the metal within the heat affected zone undergoes a significant thermal transient which can result in changes of the microstructure of the material. Such changes can significantly affect the strength, ductility and resistance to corrosion of the material in this area. The object of this study was to develop and verify a model of the gas metal arc process which would predict the thermal transient such that the mechanical and metallurgical properties of the final base metal could be predicted.

The equation of heat flow in two dimensions was modified by inclusion of terms to account for radiation and convection heat loss. In addition, the fact that material properties are a function of temperature was included. The resulting equation was highly nonlinear, but successfully approximated through use of finite difference approximations. The interior boundary of the temperature field was the boundary of the molten weld for each set of welding conditions. An empirical correlation was developed, therefore, to predict the size of the weld pool as a function of welding conditions. The model was verified by actually measuring the temperatures with thermocouples during welding.

The predicted and measured temperatures were compared and found to be in excellent agreement whenever the weld bead was truly two dimensional. Peak temperatures were predicted within an accuracy of 5% or approximately 132° F.

Thesis Supervisor: Koichi Masubuchi
Title: Professor of Ocean Engineering

ACKNOWLEDGEMENTS

First the author would like to thank Professor Koichi Masubuchi for his interest and considerable investment of valuable time.

The author also is indebted to Mr. Anthony Zona for his advice, encouragement and technical help with the experimental work.

Finally the author would like to thank Mrs. Jack "Sam" Price for her excellent typing of the final draft.

TABLE OF CONTENTS

CONTENTS

LIST OF SYMBOLS

C	Constant
c	Specific heat (Btu/lbm ° R.)
E	Voltage (volts)
H_c	Convective heat loss coefficient
H_m	Heat content (Btu/ft^3)
H_r	Radiative heat loss coefficient
I	Arc current (ampere)
$K_o K_1$	Bessel function
k	Thermal conductivity (Btu/hr ft ° R.)
N^*	Welding parameter
$\dot{Q}$	Heat flow rate (Btu/hr)
q_D	Distributed heat
r	Radial distance from arc axis (in.)
s	Time
T	Temperature (° R.)
t	Thickness (in.)
V	Speed of arc travel (ft/hr), (in/min)
w, y	Moving coordinates
w_m	Puddle length (in.)
x_o	Normalizing distance
y_m	Puddle width (in.)
α	Thermal diffusivity (ft^2/hr)
α_{min}	Minimum thermal diffusivity (ft^2/hr)
η	Welding efficiency

SYMBOLS

λ	Diffusivity factor (hr/ft^2)
ρ	Density (lb_m/ft^3)

SUBSCRIPTS:

i,j	Two-dimensional array indices
o	At ambient temperature } when used with T
m	At melting temperature } when used with T

SUPERSCRIPTS

*	Normalized quantity
-	Average value

I. INTRODUCTION

It was the objective of this thesis to develop a model of the gas metal arc welding process such that the temperature history of the metal could be predicted. The gas metal arc process was chosen because it is a type of arc welding that is sophisticated enough to be used to join complex, sensitive alloys. In addition, because high deposition rates are possible, and automatic welding machines are available, the process is compatible with production welding.

Since simple models were already in existence which work well far from the weld centerline, the primary area of concern was the so-called heat affected zone. This is the part of the base plate which, although it does not become molten, has been heated to a high enough temperature to undergo significant detectable structural change. The final goal is to be able to predict the temperature history and therefore the cool-down rate and the final microstructure in the heat affected zone as a function of welding parameters. This information would be extremely useful in development of welding processes for new high strength steels such as HY-130.

Development of the model was similar to that detailed in references 5 and 7. First a pool size and shape correlation was developed by blowing out the weld puddles with high pressure argon and measuring the resulting holes. This information was plotted as a function of nondimensional parameters and an empirical relation derived. Second, the actual temperatures were measured during welding operations. These values were compared with the results of the computer program. The input to this program

includes welding conditions and the size of the weld pool. The experimental and computed results were compared and found to be in reasonable agreement.

II. BACKGROUND

The first real success in analytical prediction of temperature histories after welding was made by Rosenthal in the late 1930's. He obtained solutions for the governing heat flow equation for situations where the source is modeled as a point, line or plane in a quasi-steady state condition. Quasi-steady state here implies that the temperature distribution is measured relative to a moving coordinate system which has its origin directly under the electrode. His solutions were also subject to the following assumptions:

(1) The physical and thermal properties of the medium are constant.

(2) The radiative heat losses from the surface to the surroundings are small.

(3) Any change of phase in the base metal and weld metal can be ignored.

(4) As was mentioned, and most important of all, the heat source is modeled as a point, line or plane source.

Experimental tests by Rosenthal and Schmerber in 1938 verified that the solutions found by Rosenthal were good sufficiently far from the weld centerline but broke down close to the weld since the molten puddle certainly has dimensions. Rosenthal's work is summarized in reference 6.

In the period between 1940 and 1967 many investigators tried with varying degrees of success to improve on Rosenthal's work. In 1952 Wells (8) used the Rosenthal equations to predict the width of the weld bead. He used experimental work done by Jackson and Shrubstall to verify his conclusions. In 1955 Apps and Milner (2) did some work in confirmation

of the work done by Wells, but even more important for purposes of this work, they first expressed the theory that the total heat input to the base plate is composed of heat from the electric arc and heat from gaseous conduction from the arc column. They were also the first to use high pressure gas to blow out the weld pools in an attempt to prove or disprove theoretical size and shape predictions. Wide variance between theory and experimental shapes was found. Adams (1) developed engineering relationships which reasonably accurately predict centerline cooling rate and peak temperatures in 1958.

One of the major obstacles to the Rosenthal solutions is that all require the heat input to the metal to be known. It is, of course, easy to measure voltage and current at the welding power supply, but, in addition, a value for the so-called welding efficiency must be estimated. In 1965 Christensen et al. (3) in a detailed work measured arc efficiency. Unfortunately the range of values for each welding process is very large. In addition in this work methods for finding length, width, depth, and cross sectional area of the weld bead were proposed. Christensen's suggestions for nondimensional welding parameters are also important to this work.

The importance of knowing the exact temperature histories in the heat affected zone was summarized nicely in the 1958 Adams lecture to the American Welding Society by E.F. Nippes. (4) Significant changes can occur in the microstructure in this area. As metals become increasingly sophisticated and sensitive to high heat these changes become even more important. For very high strength materials welding processes must use

the optimal heat input to do the least amount of damage if the strength of the base plate is to be preserved. For these reasons the temperature history in the heat affected zone must be known in detail so that the cool-down rate can be computed and microstructure predicted through use of a continuous cooling transformation diagram. This idea was suggested by Nippes in 1958.

The failings of all previous models have been that they are primarily based on the Rosenthal solution and its inherent approximations. As was mentioned previously these models are good far enough from the weld center-line, but become inaccurate in the heat affected zone which is the area of primary interest. It has been proven that to make accurate predictions the following must be considered:

(1) The heat source in the metal is of finite size.

(2) The effect of the variations of thermal properties of the material;

(3) radiative and convective heat loss;

(4) distributed heat.

Distributed heat is that part of the energy which is transferred to the base plate through the gas cloud. The inclusion of these factors makes the original partial differential equation highly nonlinear. It has finally become possible to approximate the solution to such problems through the use of a large digital computer.

The first truly accurate prediction of temperature in the heat affected zone was published in 1967 and 1968 by Tanbakuchi (7) and Pavelic. (5) Their method derives an empirical correlation for molten puddle size

and shape, and then uses the boundary of the molten pool as one boundary condition in a finite difference solution. Errors of less than 10% were achieved. The reason for their marked success was that not only were the factors listed above included, but their work was limited to the gas tungsten arc process, thus eliminating for the first time the dependence on an estimated welding machine efficiency.

The principles developed by Pavelic and Tanbakuchi form the basis for this work. The gas metal arc process was chosen for previously mentioned reasons. The spray transfer mode was chosen because it is compatible with the higher welding speeds and with more different types of metals. Two shielding gases were tried, argon and argon with 5% oxygen. The latter was chosen to improve the stability of the arc and to reduce weld spatter. Low carbon steel was used because its properties are well known and it is readily available. The computer model was restricted to two dimensionality to reduce the computer memory required, which is very large even for the two-dimensional case.

III. POOL SIZE AND SHAPE CORRELATION

Since any sort of finite difference algorithm requires accurate boundary conditions this aspect of the project was very important, and had as its goal the derivation of empirical relationships which would predict points on the melting isotherm of the weld puddle. Then, since the melting temperature of the weld and the ambient temperature are known, the required number of boundary conditions are known. A theoretical analysis is stated in some detail in reference 7. The conclusion reached in his work was that an analytical method of determining the points of maximum length and width is possible. However, this analytic solution is tied to the 2D solution of the heat conduction equation and is not confirmed by experiment. As a result, in an attempt to profit from previous experience detailed in reference 5, the blow-out method of determining size and shape of the puddle was tried first.

A. Experimental Procedure

All experimental work was done on a LINDE automatic gas metal arc system fed by a LINDE type SVI 300 welding power supply. After considerable trial and error a satisfactory blow-out system was constructed and is depicted schematically in Figure 1. The copper tubing was attached to the torch of the welding machine so that it had a constant orientation with respect to the arc and the molten puddle. The flow restricting valve was a coarse regulating needle valve and was left partially open. The blowing out of the puddle was accomplished by stopping the welding machine and opening the gate valve simultaneously. This apparatus was simple and unsophisticated but was found to be effective once sufficient experience

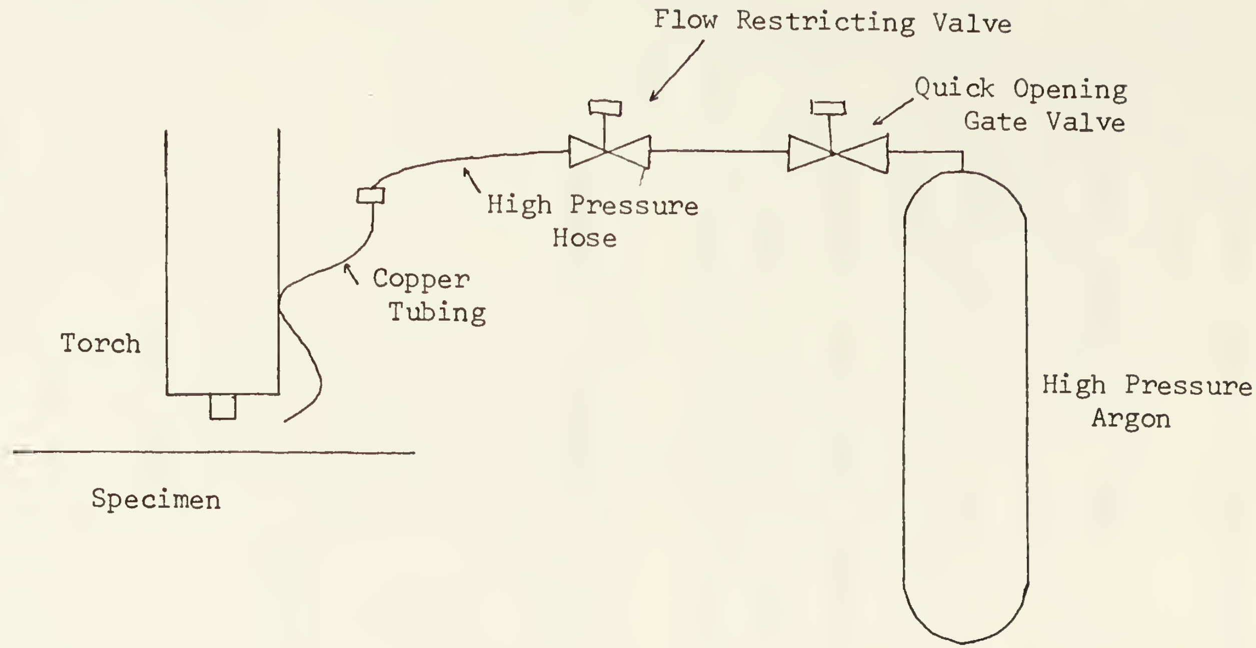


Figure 1. PUDDLE BLOW-OUT SYSTEM

running the welding machine had been accumulated and once the correct angle of the final length of copper tubing was determined. The pressure of the argon used to blow the puddles out was not measured but is estimated to be 500 pounds per square inch.

All the blow-out specimens measured two inches by 10.8 inches. This size specimen was chosen because it was narrow enough to be almost free of distortion during the welding and yet wide enough to prevent heat reflected at the edges from affecting the welding conditions. Experience also showed the particular welding machine used consistently produced a stable weld after an initial run of three to four inches. Therefore a length of over ten inches allowed some choice of the exact moment when the puddle was blown out.

The following table summarizes the welding conditions for the blow-out specimens.

Voltage	15-20 volts
Current	190-280 amps
Speed	28.5 to 48.7 inches per minute
Thickness	1/16th inch
Material	mild steel

The driving parameter is the speed. It was found that at each new speed setting the voltage and current must be adjusted very exactly to get a two-dimensional puddle and yet to prevent the puddle from dripping through. Attempts were also made to get two-dimensional puddles in 1/8th inch steel, but the increased width of the weld puddle allowed the molten material to

drip through before it could be blown out. This result is in agreement with Pavelic's. Experimental results are included in Appendix C.

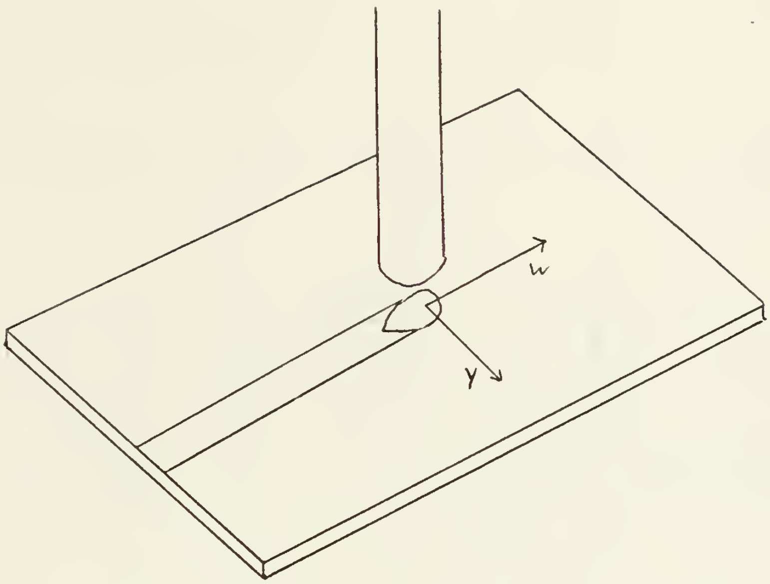
B. Empirical Correlation

By manipulation of the line source solution to the heat conduction equation a nondimensional welding parameter can be derived and is defined as follows:

$$N^* = \frac{3.415\ EI}{2\pi\bar{\alpha}tH_m}$$

Details are included in Appendix D. This parameter is a function of the material properties and of the welding parameters of primary interest, namely voltage and current. It is not so obvious that the above parameter provides a method of excluding welding efficiency from further consideration for this procedure. If the dimensions of the puddle can now be expressed in terms of N^*, then for a particular machine, the isotherm of melting can be determined and the remainder of the solution is a standard finite element procedure consisting of averaging the temperatures of surrounding nodes until the solution converges.

There is one flaw in this reasoning, however, and that is that the effect of the welding speed has not been included. A coordinate system is defined as follows:



where w is the moving coordinate in the direction of the welding and y is the coordinate used to express distance from the weld centerline. These coordinates can be nondimensionalized through dimensional analysis as follows:

$$y^* = \bar{\lambda} V y$$

$$w^* = \bar{\lambda} V w$$

and the effect of welding speed has been included.

In a manner similar to that used by Pavelic the blown-out weld puddles were measured using a Nikon machinist's microscope. These results were fed into the service program "puddle fit" which is included in Appendix B. As can be seen from Figures 2 and 3 the data falls roughly into an area which might be described by a relationship of the form

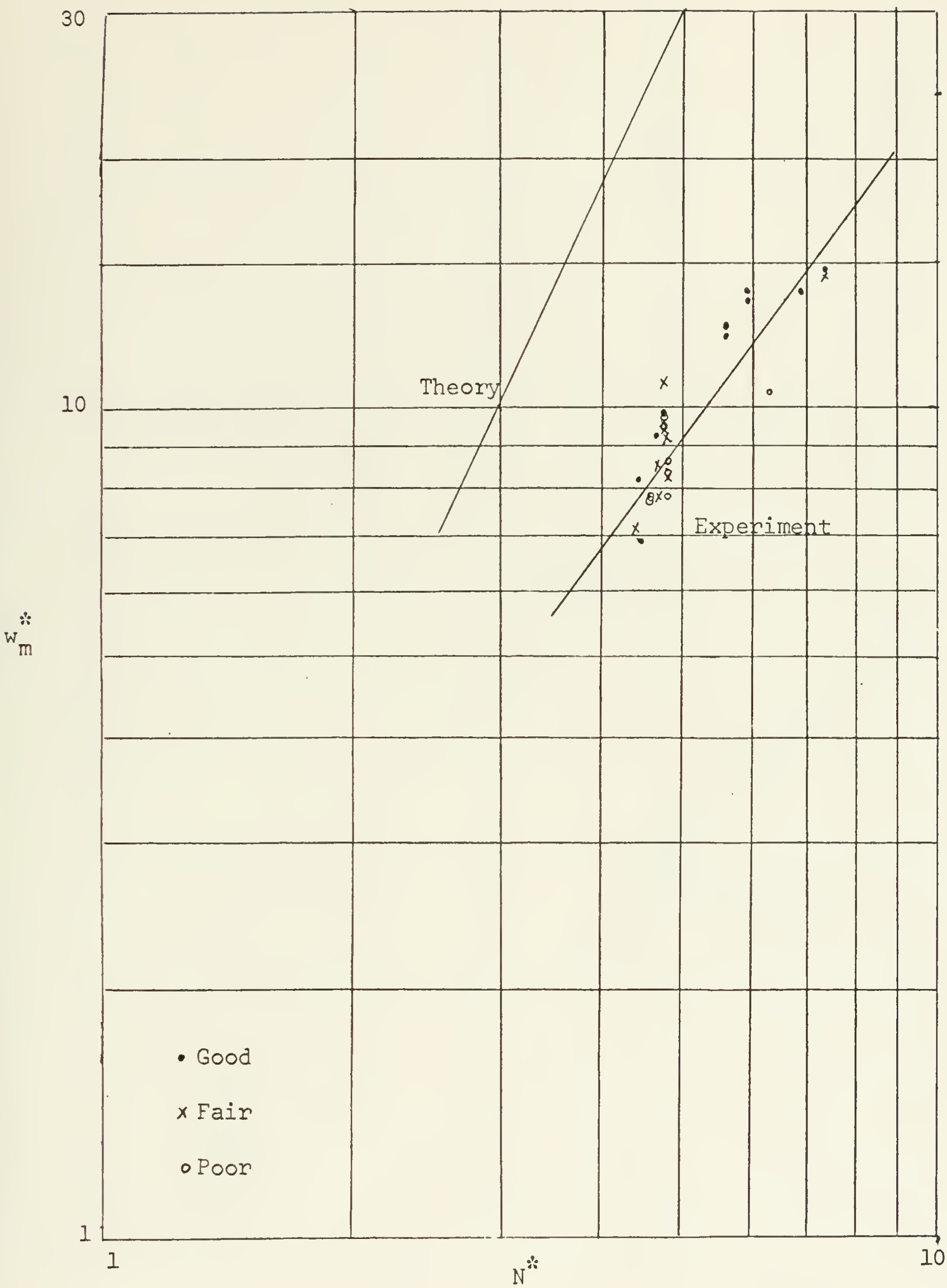


Figure 2. MAXIMUM PUDDLE LENGTH CORRELATION

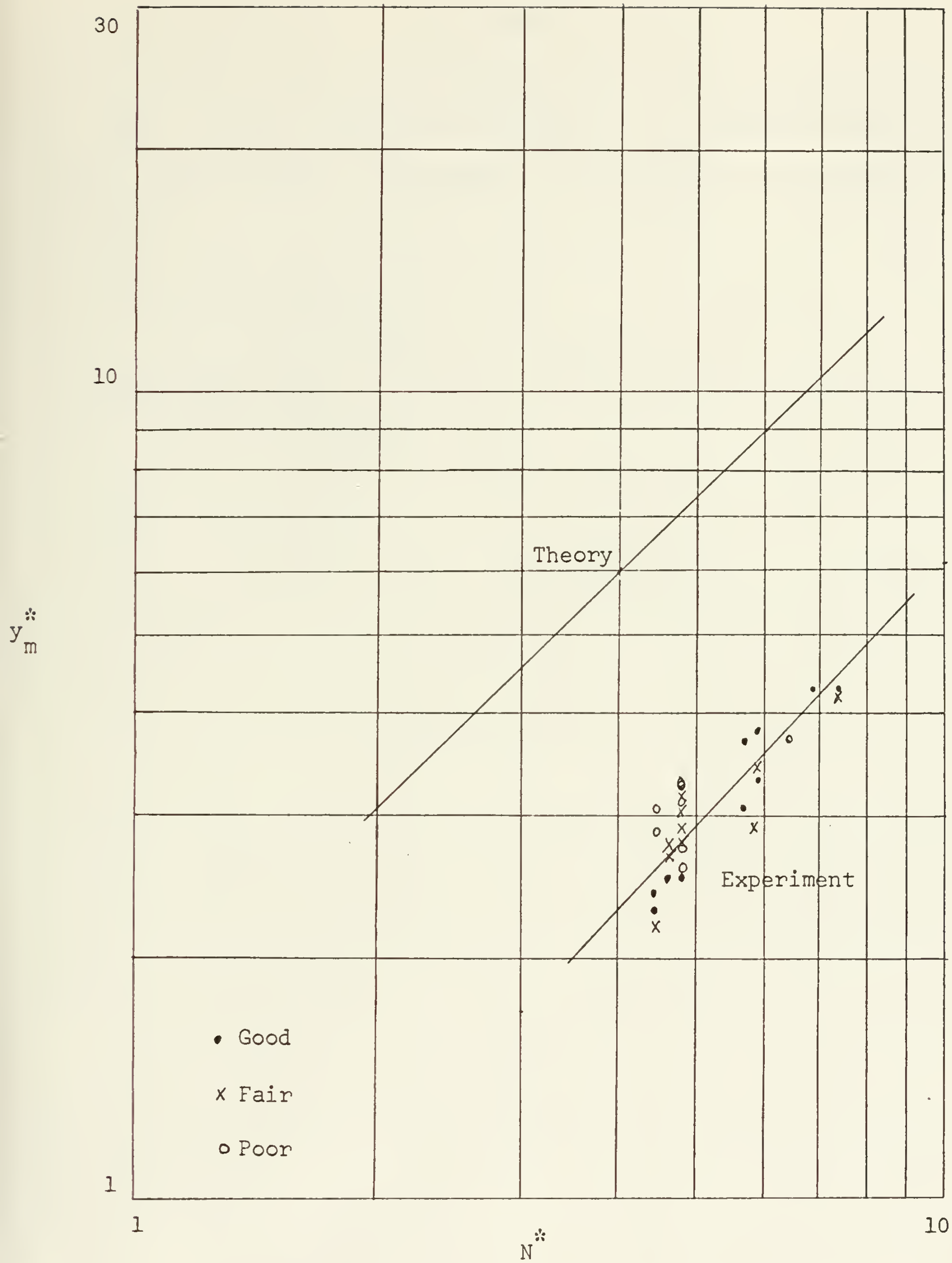


Figure 3. MAXIMUM PUDDLE WIDTH CORRELATION

$$w^* = CN^{*m}$$

where C is the intercept of a straight line on a log log plot and m is its slope. Program "puddle fit" performed this operation using a least squares approximation. The results of this program are the following relations:

for maximum length

$$w_m^* = 1.048\ N^{*1.359}$$

for maximum width

$$y_m^* = .549\ N^{*1.041}$$

IV. REVIEW OF THE COMPUTER MODEL

This section is a review of the computer model and a discussion of the minor changes made to it. The basic equations are unchanged. That is the model retains correction terms for diffused heat, convection, radiation, and variable material properties. Changes were of a minor sort and were intended to, first of all, make the program run in a speed range higher than that considered previously and second, to reduce the actual run time of the entire program.

A. Basic Equations Used

If it is assumed there is no heat generation in an isotropic solid and that the thermal properties are a function of temperature, the differential equation of heat may be written

$$\nabla \cdot (k\nabla T) \cdot \rho c \frac{DT}{Ds} \qquad (4\text{-}1)$$

for a standard Cartesian coordinate system. If a quasi-stationary state is defined in which temperature is not a function of time and the coordinate system is measured relative to the moving arc this equation is changed to the following:

$$\nabla \cdot (k\ T) = -\rho c\ V \frac{\partial T}{\partial w} \qquad (4\text{-}2)$$

where w is the coordinate in the direction of motion and is described by the following:

$$w = x - Vs$$

If a normalizing factor is defined

$$x_o = \frac{S\ \alpha_{min}}{V} \qquad (4\text{-}3)$$

where S is a constant and will be discussed in detail later. Then all dimensions can be normalized by dividing by x_o. Similarly temperatures are normalized by dividing by the difference between melting temperature and ambient temperature, that is,

$$T^* = \frac{T - To}{Tm - To}$$

After normalization, rearrangement, and introduction of the two-dimensional approximation equation (4-2) becomes:

$$\frac{\partial^2 T^*}{\partial w^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} + \frac{V\ x_o}{\alpha(T^*)}\frac{\partial T^*}{\partial w^*} + \frac{1}{k(T^*)}\frac{\partial k(T^*)}{\partial T^*}\left[\left(\frac{\partial T^*}{\partial w^*}\right)^2 + \left(\frac{\partial T^*}{\partial y^*}\right)^2\right] = 0 \qquad (4\text{-}4)$$

However, as was clearly shown by the differences in reference 5 and 7, not all the energy is transferred into the base plate through the molten pool. Some energy is transferred directly from the ionized shielding gas to the base plate. A function which approximates this distributed heat is given below:

$$q_D(r) = q\ (0)e^{-Cr^2}$$

where $r^2 = w^2 + y^2$

and which when normalized becomes

$$q_D(w^*, y^*) = \frac{q(0)}{(T_m - T_o)}\ e^{-Cx_o^2(w^{*2} + y^{*2})}$$

Losses due to convection and radiation can also be included and the result is:

$$\frac{\partial^2 T^*}{\partial w^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} + \frac{x_o^2}{tk(T^*)}\Big\{q_D(w^*, y^*) - H_c T^*$$

$$-H_r\left([T^*(T_m - T_o) + T_o]^4 - T_o^4\right)\Big\} + \frac{V\ x_o}{\alpha(T^*)}\ \frac{\partial T^*}{\partial w^*}$$

$$+ \frac{1}{k(T^*)} \frac{\partial k(T^*)}{\partial T^*} \left\{ \left(\frac{\partial T^*}{\partial w^*} \right)^2 + \left(\frac{\partial T^*}{\partial y^*} \right)^2 \right\} = 0 \tag{4-5}$$

In accordance with standard procedures the partial differentials are now replaced by finite differences which are now defined.

$$\frac{\partial T^*}{\partial w^*} \simeq \frac{(a_{i,j} - a_{i-1,j})}{a_{i,j} a_{i-1,j}} T^*_{i,j} + \frac{a_{i-1,j}}{a_{i,j}\,(a_{i,j} + a_{i-1,j})} T^*_{i+1,j} -$$

$$\frac{a_{i,j}}{a_{i-1,j}(a_{i,j} - a_{i-1,\,j})} T^*_{i-1,j}$$

$$\frac{\partial T^*}{\partial y^*} \simeq \frac{b_{i,j} - b_{i-1,j}}{b_{i,j}\; b_{i,j-1}} T^*_{i,j} + \frac{b_{i,j-1}}{b_{i,j}\,(b_{i,j} + b_{i,j-1})} T^*_{i,j+1} -$$

$$\frac{b_{i,j}}{b_{i,j-1}\,(b_{i,j} + b_{i,j-1})} T^*_{i,j-1}$$

$$\frac{\partial^2 T^*}{\partial w^{*2}} \simeq - \frac{2}{a_{i,j}\; a_{i-1,j}} T^*_{i,j} + \frac{2}{a_{i,j}\,(a_{i,j} + a_{i-1,j})} T^*_{i+1,j} +$$

$$\frac{2}{a_{i-1,j}\,(a_{i,j} + a_{i-1,j})} T^*_{i-1,j}$$

$$\frac{\partial^2 T^*}{\partial y^{*2}} \simeq - \frac{2}{b_{i,j}\; b_{i,j-1}} T^*_{i,j} + \frac{2}{b_{i,j}\,(b_{i,j} + b_{i,j-1})} T^*_{i,j+1} +$$

$$\frac{2}{b_{i,j-1}\,(b_{i,j} + b_{i,j-1})} T^*_{i,j-1}$$

where the computational module is defined in Figure 4.

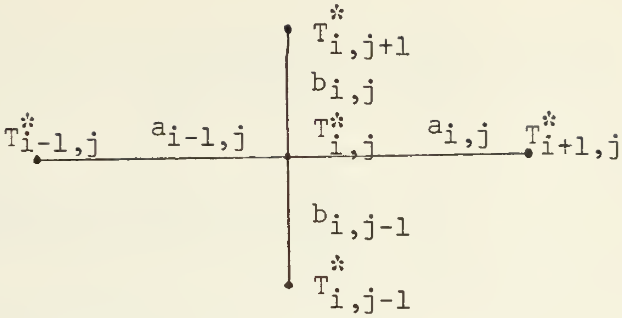


Figure 4.

Substitution of these expressions into equations and rearranging yields the following:

$$T^*_{i,j}\left[-\frac{2}{a_{i,j}\,a_{i-1,j}} - \frac{2}{b_{i,j}\,b_{i,j-1}} + \frac{1}{\alpha(T^*_{i,j})}\;\frac{a_{i,j}-a_{i-1,j}}{a_{i,j}\,a_{i-1,j}} - \frac{BH_c}{k(T^*_{i,j})}\right] + T^*_{i+1,j}\left[\frac{2\alpha(T^*_{i,j})+a_{i-1,j}}{\alpha(T^*_{i,j})\,a_{i,j}\,(a_{i,j}+a_{i-1,j})}\right] +$$

$$T^*_{i-1,j}\left[\frac{2\,\alpha(T^*_{i,j})-a_{i,j}}{\alpha(T^*_{i,j})\,a_{i-1,j}\,(a_{i,j}+a_{i-1,j})}\right]$$

$$T^*_{i,j+1}\left[\frac{2}{b_{i,j}\,(b_{i,j}+b_{i,j-1})}\right] + T^*_{i,j-1}\left[\frac{2}{b_{i,j-1}\,(b_{i,j}+b_{i,j-1})}\right]$$

$$+\frac{B}{k(T^*_{i,j})}\left[\frac{q(0)}{T_m-T_o}\,e^{-Cx_o^2}(w^{*2}+y^{*2}) - H_r\left\{[T^*_{i,j}(T_m-T_o)+T_o]^4 - T_o^4\right\}\right] +$$

$$\frac{1}{k(T^*_{i,j})}\;\frac{\partial k(T^*)}{\partial T^*}\left[\frac{a_{i,j}-a_{i-1,j}}{a_{i,j}\,a_{i-1,j}}\,T^*_{i,j} + \frac{a_{i-1,j}}{a_{i,j}(a_{i,j}+a_{i-1,j})}\,T^*_{i+1,j} - \frac{a_{i,j}}{a_{i-1,j}\,(a_{i,j}+a_{i-1,j})}\,T^*_{i-1,j}\right\}^2 +$$

$$\left\{ \frac{b_{i,j} - b_{i,j-1}}{b_{i,j}\, b_{i,j-1}} T^*_{i,j} + \frac{b_{i,j-1}}{b_{i,j}\,(b_{i,j} + b_{i,j-1})} T^*_{i,j+1} - \frac{b_{i,j}}{b_{i,j-1}\,(b_{i,j} + b_{i,j-1})} T^*_{i,j-1} \right\} \Bigg]^2 = 0$$

B. Convergence and Uniqueness

In reference 7 Tanbakuchi used the following criteria for grid spacing parameters in order to insure convergence:

$$h_{max} < \frac{2\alpha_{min}}{\alpha_o}$$

If h_{max}, which is a nondimensional grid spacing parameter in the program itself, is less than twice the minimum value of diffusivity divided by another value of diffusivity, then the solution will converge and will be unique. If h_{max} is arbitrarily set to 1, then

$$\alpha_o = 2\alpha_{min}$$

and convergence and uniqueness will be satisfied if the length normalizing factor is defined

$$x_o = \frac{2\alpha_{min}}{V} \qquad (4\text{-}7)$$

However, it was found that in the speed range greater than approximately 30 inches per minute the length normalizing factor became very small which caused the puddle dimension to become very large. As a result, computer limits for division by very small numbers were exceeded in subroutines TYPFN or COLUMN because k_o bessel function values returned from subroutine BESSEL were very small. Similar problems had to be corrected

throughout the program as a result of the increased welding speed. The solution to this problem was to increase the length normalizing factor, x_o, by replacing the 2 in equation (4-7) by an arbitrary factor (see equation 4-3). This factor can be set at a value higher than 2 if the grid spacing parameter is set proportionally lower than 2. For most runs S equaled 4 and h_{max} equaled .5.

Unfortunately, however, such careful work does not insure convergence. As is pointed out by Pavelic, equation (4-4) is highly nonlinear and therefore a finite difference approximation need never converge. The experimental results are available, however, for ready comparison so that a diverging solution should be immediately obvious.

C. <u>Changes in the Computer Program</u>

As was mentioned at the beginning of this section, the changes made to the program were minor, but worth mentioning. The first change was in the definition of the normalization factor and has been discussed in some detail. The second change was the establishment of two holding arrays in which successive iterations of the temperature surface are stored. After each complete iteration each element in each array is compared with the corresponding element in the other. If every element is within a tolerance which can be specified in the input deck, execution stops and results are printed out. This improvement in the program reduced run time on an IBM 360 model 155 from several minutes to approximately one-and-one-half minutes. However, all the changes made so little difference to the structure of the program that the flow charts in reference 7 still apply.

V. EXPERIMENTAL MEASUREMENT OF TEMPERATURE HISTORY

Experimental temperatures were measured using the same welding apparatus described in section 3. Specimens were either 1/16th inch or 1/8th inch cold rolled SAE 1018 steel and measured 17.5 inches long by eight inches wide. Six thermocouples made of 24-gauge chromel and alumel wire were placed in each specimen. Filler metal was Airco type 675 which is recommended for low-carbon steel, and argon plus five percent oxygen continued to be used as the shielding gas. A Honeywell type 1508 recorder was used to measure and present the output of the thermocouples. A schematic of the measurement circuit is presented in Figure 5.

A. Thermocouple Emplacement

Welding conditions were first established using scrap steel. Then program puddle fit was used to estimate the width of the weld bead. Then thermocouples were placed at varying distances from the weld centerline, starting just outside the predicted width of the weld bead. Since temperatures in the base plate can be predicted accurately by the point source theory as close as 1/4 of an inch from the weld centerline, all thermocouples on all three temperature measuring runs were placed at distances less than .25 inches from the weld centerline.

Since 24-gauge wire is approximately .020 inches in diameter, thermocouple holes were drilled with a diameter of approximately .040 inches. After the holes were drilled the wire was faced through the plate and the ends of the two wires were melted together. The small bead thus formed was hammered into the top surface of the plate to insure an intimate contact with the entire surface of the hole. After the thermocouples were in

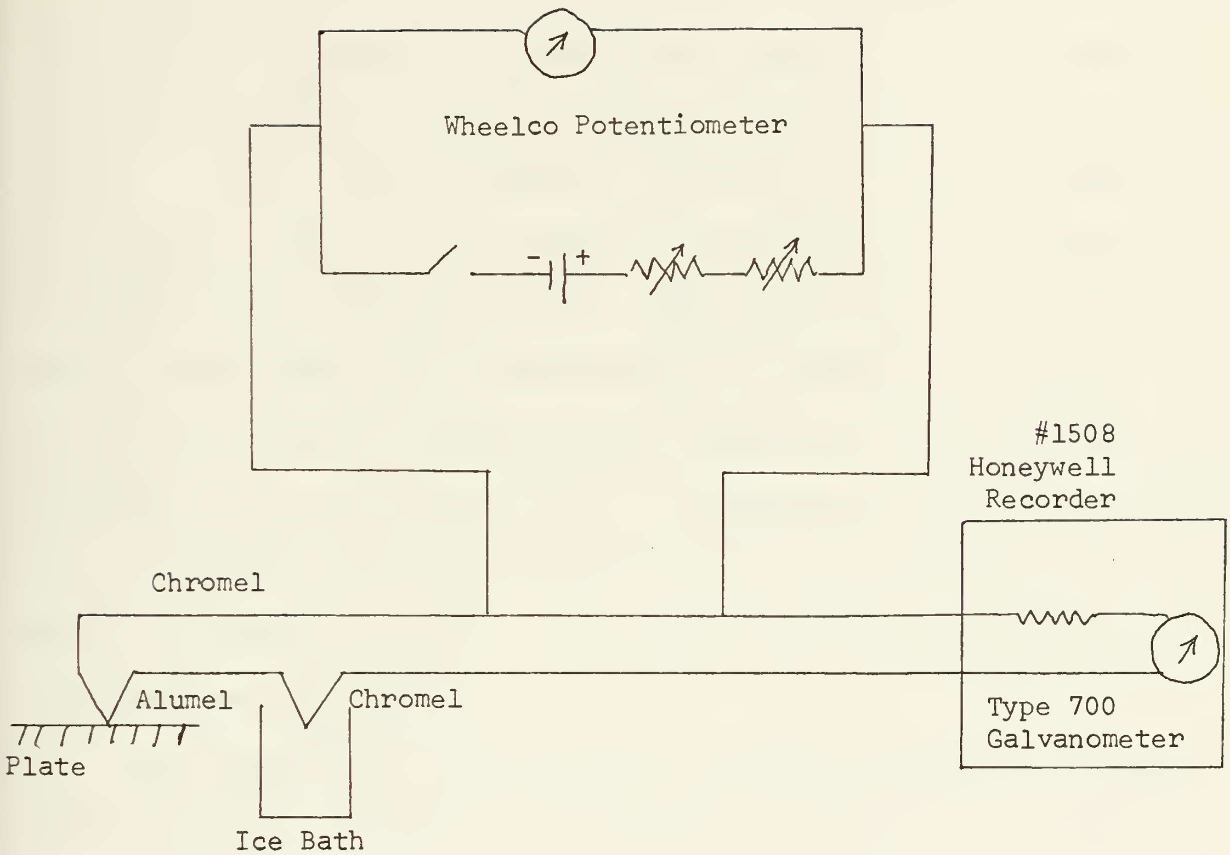


Figure 5. THERMOCOUPLE CIRCUIT

place the actual distance from the weld centerline to the centerline was measured using the microscope mentioned previously. Figure 6 depicts a typical thermocouple installation.

B. Temperature Measurement Procedure

Figure 7 depicts the welding machine during a temperature measurement run. Voltage and current were measured by meters on the power supply. Speed was measured by using a stop watch to time travel over a known distance. To start a run the Honeywell recorder was started at a speed of one inch per second. It is a feature of this recorder that timing marks are automatically made on the paper. Next the welding machine was started, and the start point of the welding machine was noted on the recorder paper. The welding machine was stopped several inches after the last thermocouple was passed, and the recorder after all temperatures had decayed sufficiently. In addition, for program "experimental data," the distance from the start of the weld to the first thermocouple and the distances between thermocouples were measured.

Three experimental runs were made with varying degrees of success. Welding parameters for each are listed below in Table 1.

TABLE 1.

Run	1	2	3
Voltage (volts)	15	16.5	22
Amperage (amps)	190	235	325
Thickness (in.)	1/16	1/16	1/8
Speed (ipm)	28.5	43.5	28.8

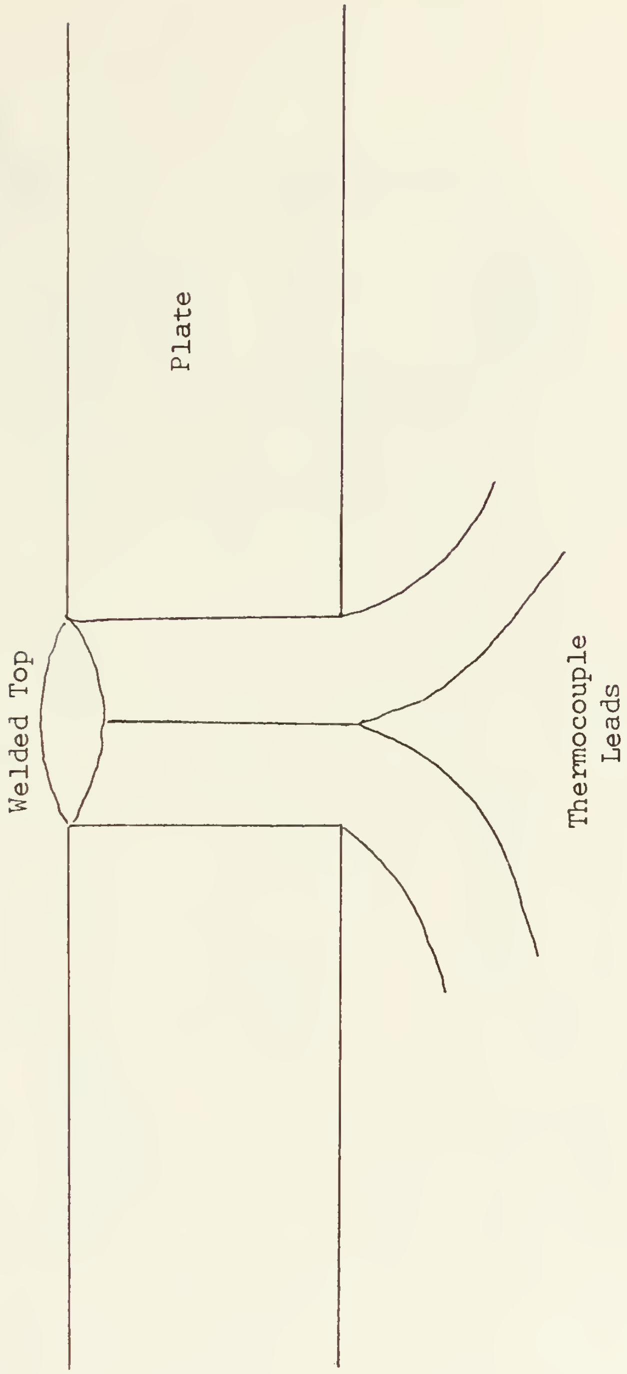


Figure 6. THERMOCOUPLE INSTALLATION

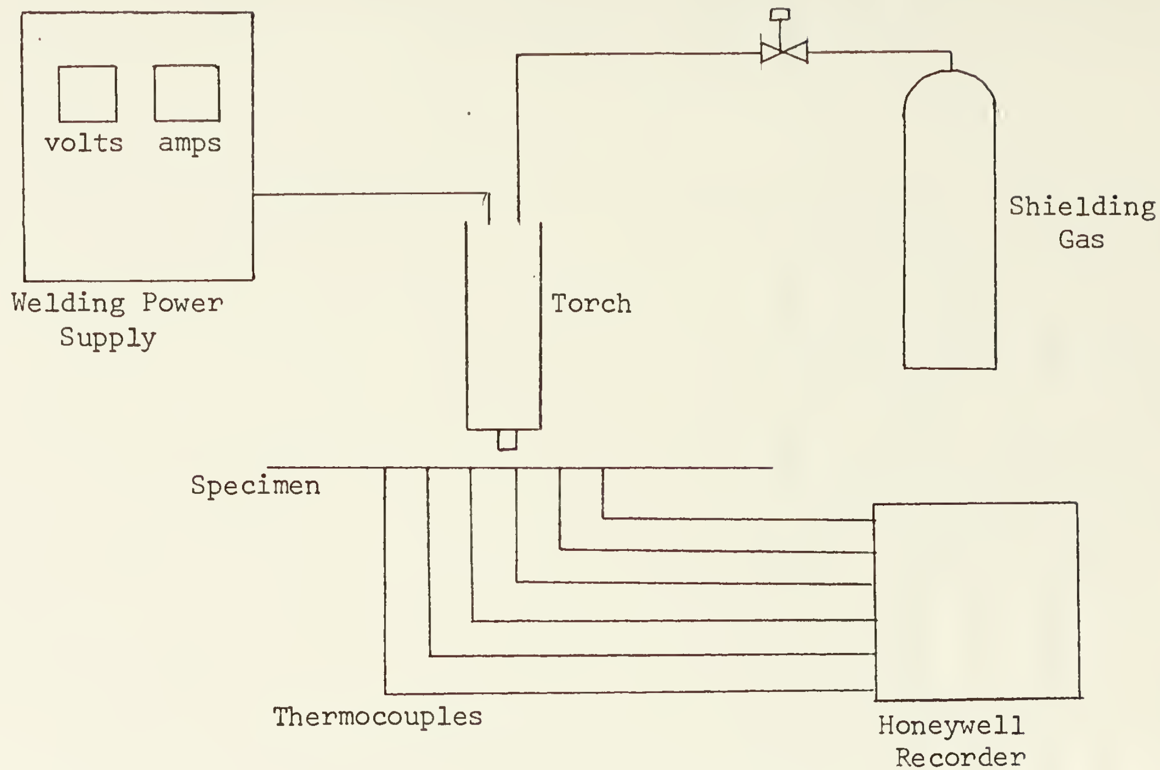


Figure 7. EXPERIMENTAL APPARATUS FOR MEASURING TEMPERATURES

C. Analysis of Experimental Data

The output of the Honeywell recorder is in the form of deflections continuously displayed. A sample curve is reproduced in Figure 8. The deflection was measured every half second and converted to a temperature by use of a calibration curve. These temperatures for an experimental run are included in Appendix C. This data plus the distances already mentioned form the input to program "experimental data" which puts the experimental data in the same coordinate system as the computer output for comparison. A listing of program experimental data is contained in Appendix B.

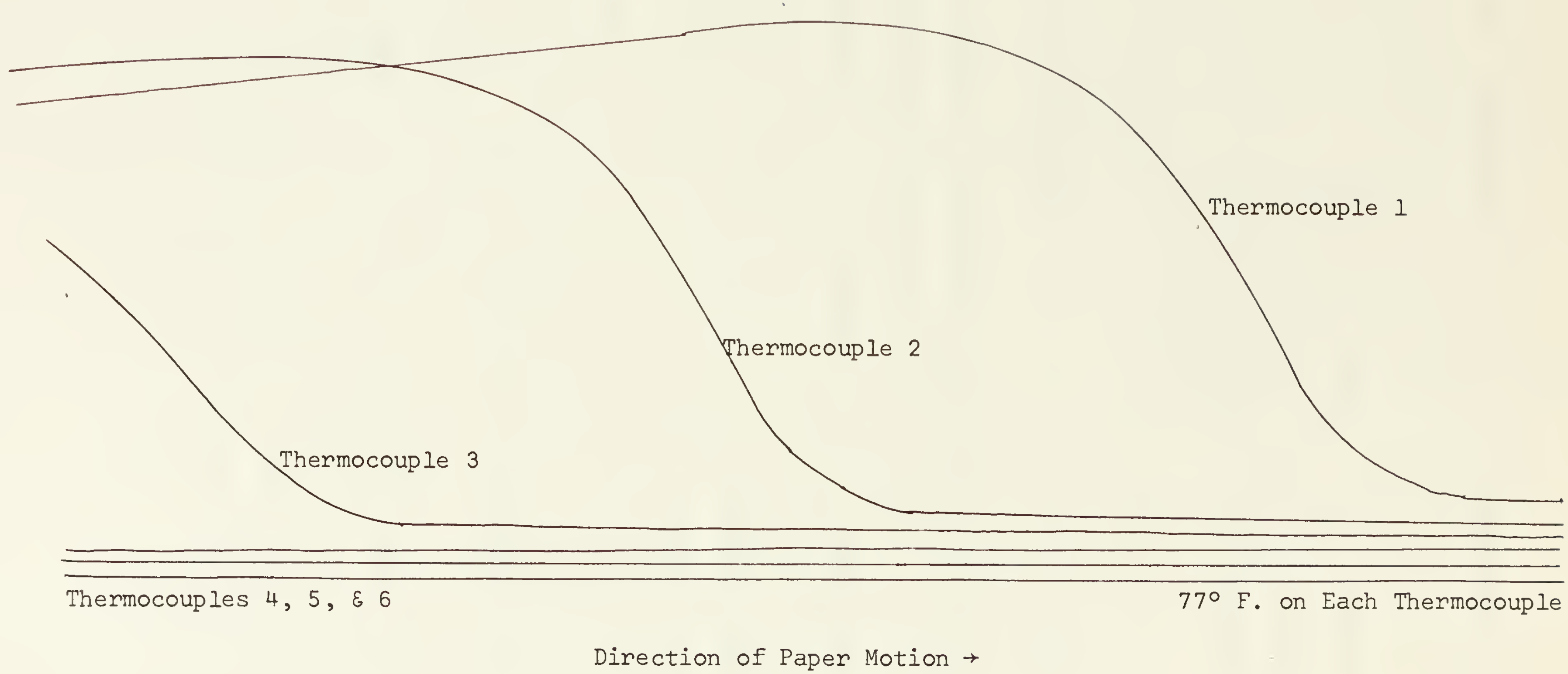


Figure 8. REPRESENTATION OF THERMOCOUPLE OUTPUTS ON THE HONEYWELL RECORDER

VI. COMPARISON OF COMPUTER AND EXPERIMENTAL RESULTS

Welding parameters for the three experimental runs are listed in Table 1 in section V. Figures nine through 15 are a comparison of the temperatures measured in runs 1 and 3 with those computed using the welding parameters and the thermocouple location. For information the thermocouple locations are included in Appendix C.

If error is defined:

$$\text{ERROR} = \frac{Tc^{*} - Te^{*}}{1.0} \qquad (100\%)$$

then the error ranges in magnitude from as little as five percent at the maximum separation for thermocouple four to as great as 18 percent at the maximum separation for thermocouple one. This represents a difference in temperature ranging between 132° R. and 476° R. at the points of maximum separation. With this particular definition of error, the error observed is somewhat higher than that claimed by Pavelic in his work. It must be remembered, however, that his definition of error is based on a comparison of the peak temperatures regardless of when they occur relative to each other. Using his definition the magnitude of the observed error is decreased considerably.

As Pavelic pointed out the ideal comparison between experimental and computed data would occur if the curves were to overlap. However, in practice this is difficult to accomplish. Since experimental results generally lag the computed results, the contact between the wall of the hole and the thermocouple is immediately suspect. In fact, however, the

problem may be in the numerous factors and variables which are a part of welding and yet are extremely hard to quantify.

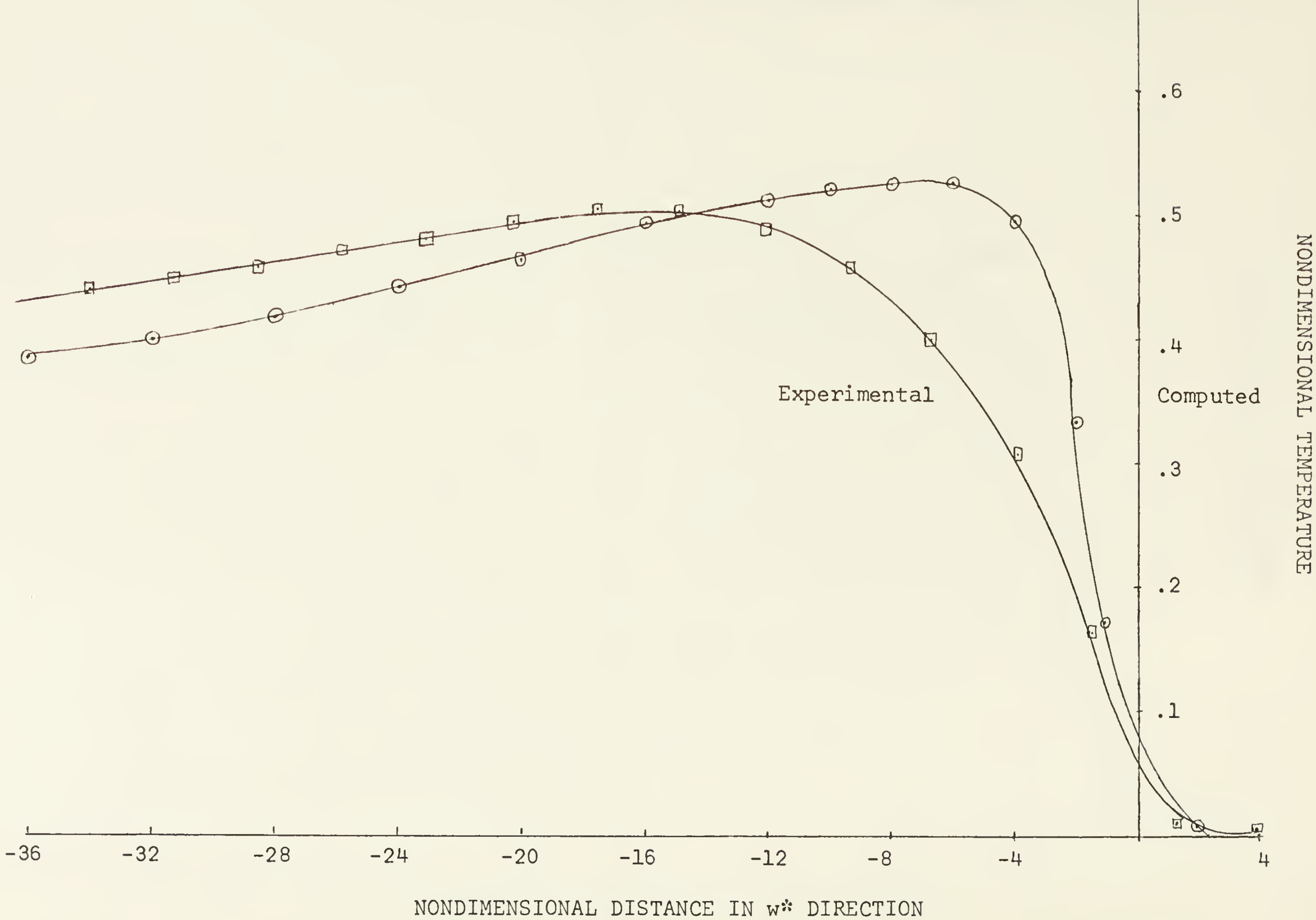
As was mentioned before, welding conditions were determined first by choice of speed. The speed forced an optimal choice of voltage and current since the desired goal was a two-dimensional weld bead. However, as the speed or thickness increased, it was observed that the surface of the molten puddle flattened out and the resulting bead departed radically from true two dimensionality. Experimental runs one and two were sectioned and macroscopically etched to determine the shape of the weld bead cross section. The results are represented in Figure 16. Since the thermocouples were a discontinuation in the specimen surface, all thermocouples came in contact with the molten metal and consequently all gave relatively high readings. Figure 15 is thermocouple three of run three. Using <u>either</u> definition, the error for this thermocouple is approximately 20 percent and is unacceptably high, and this was the best thermocouple from this run.

The problem in these two runs is that too much power is being transferred to the metal. All welding handbooks consulted advised reducing the current and the voltage to reduce spatter and to reduce the tendency of the weld bead to spread out at the top. Of course, if the power were reduced, the penetration is reduced and two dimensionality is lost anyway. One other possible variable is the type of welding gas. Argon plus 5% oxygen was used, however, which is recommended for gas metal arc welding on low-carbon steel.

The conclusion reached was that the method first used by Tanbakuchi

and Pavelic is applicable to gas metal arc welding in a spray transfer mode. However, a lesser degree of success was achieved at very high welding speeds.

Figure 9
Run #1
Thermocouple 1



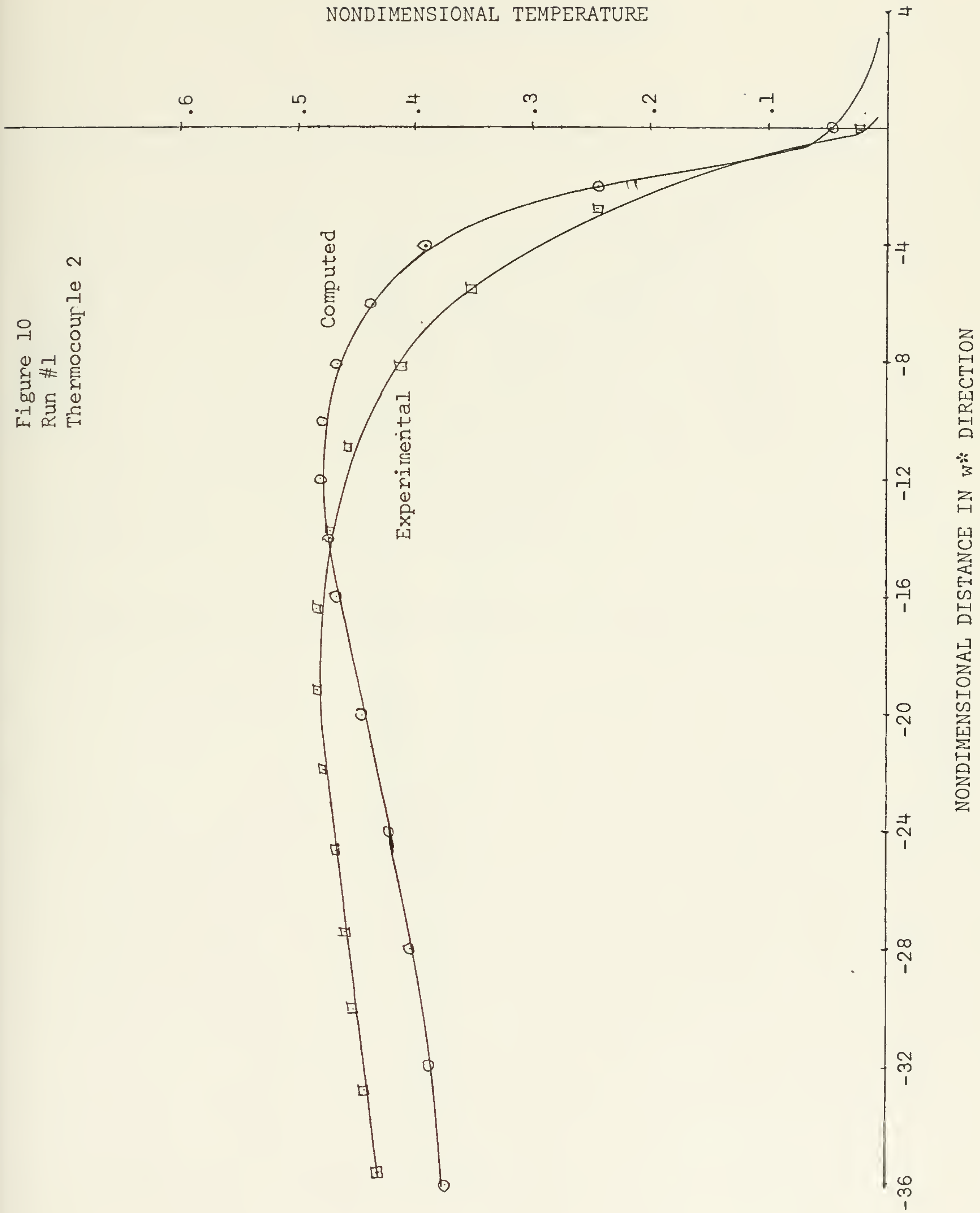


Figure 10
Run #1
Thermocouple 2

Figure 11
Run #1
Thermocouple 3

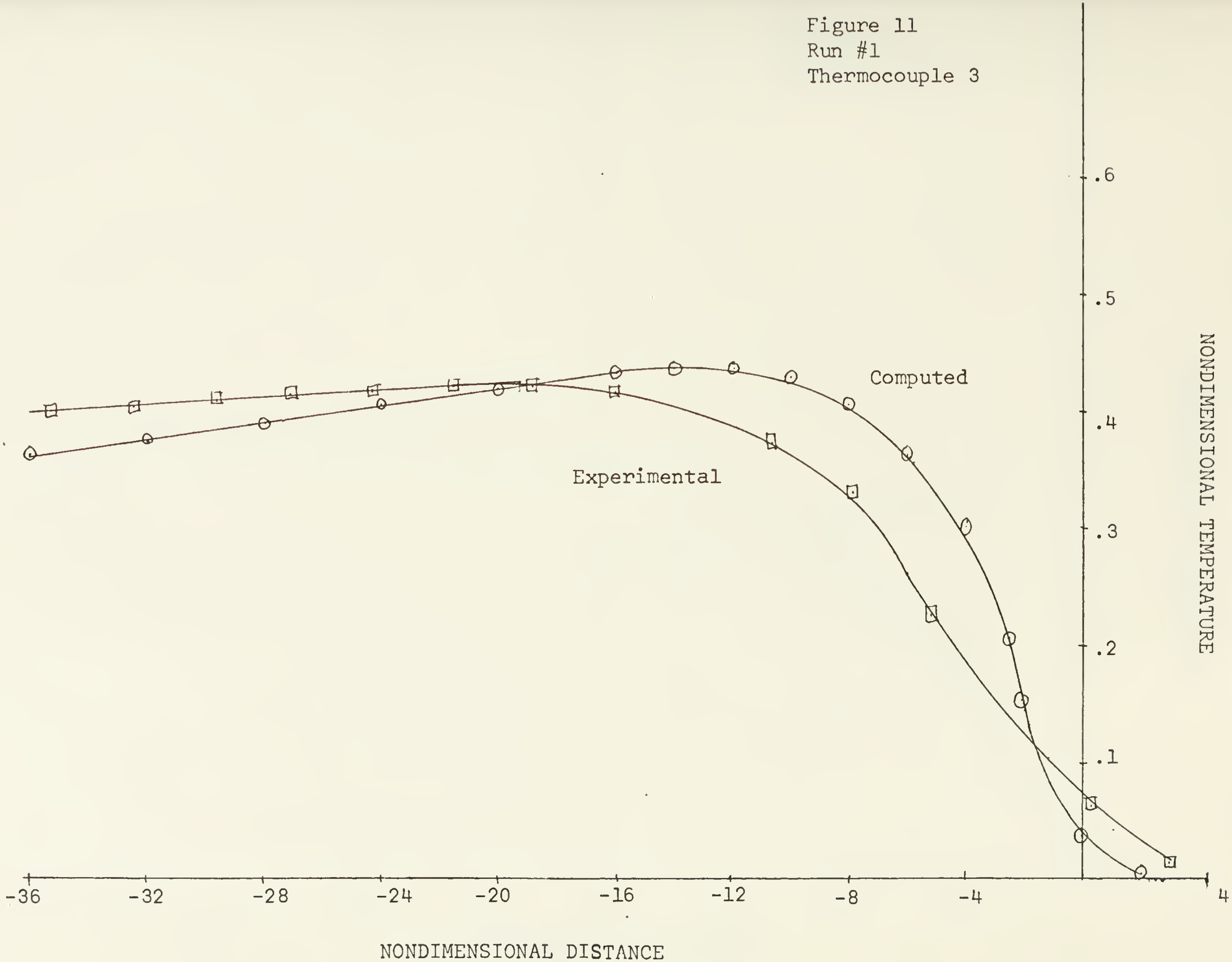
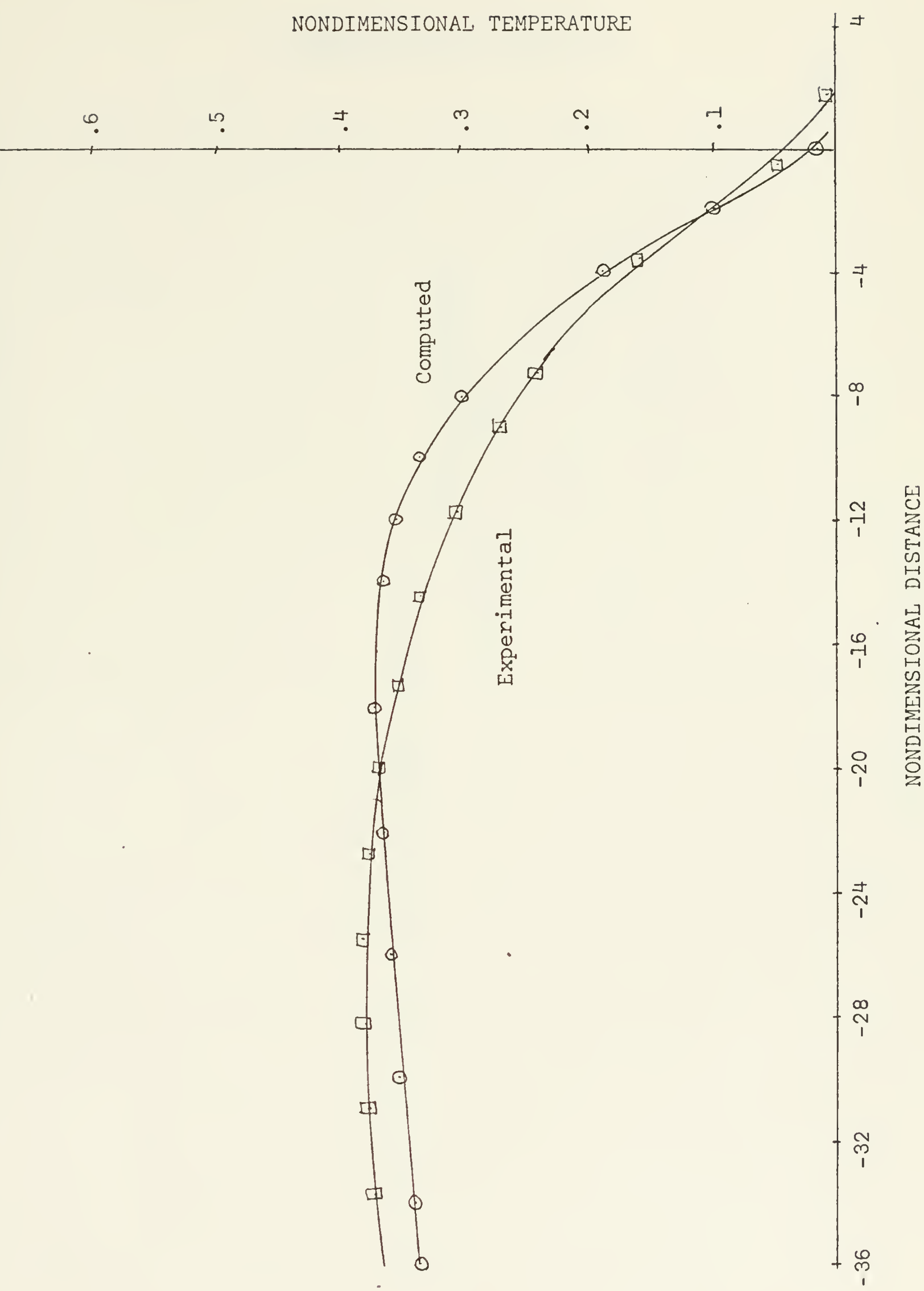


Figure 12
Run #1
Thermocouple 4



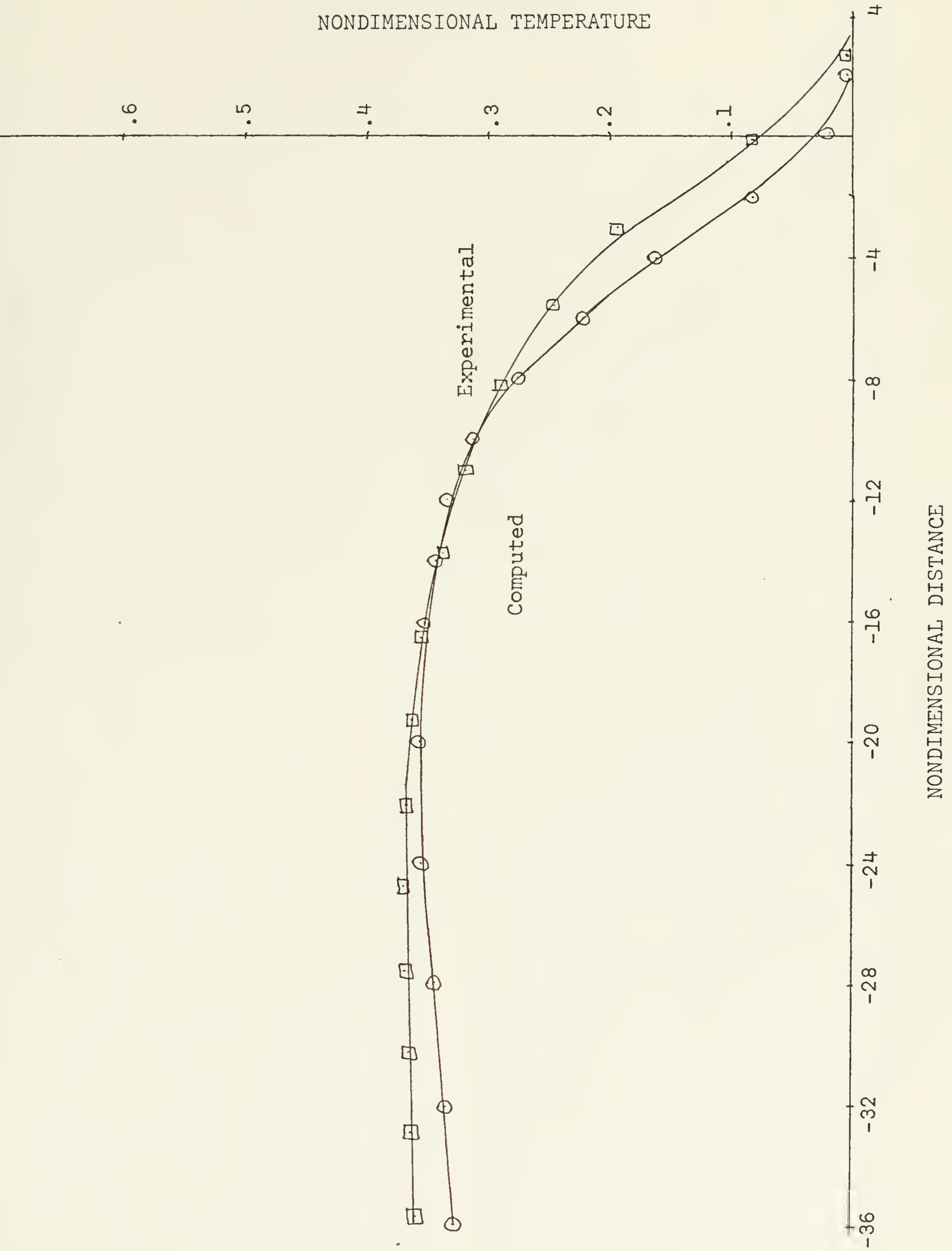
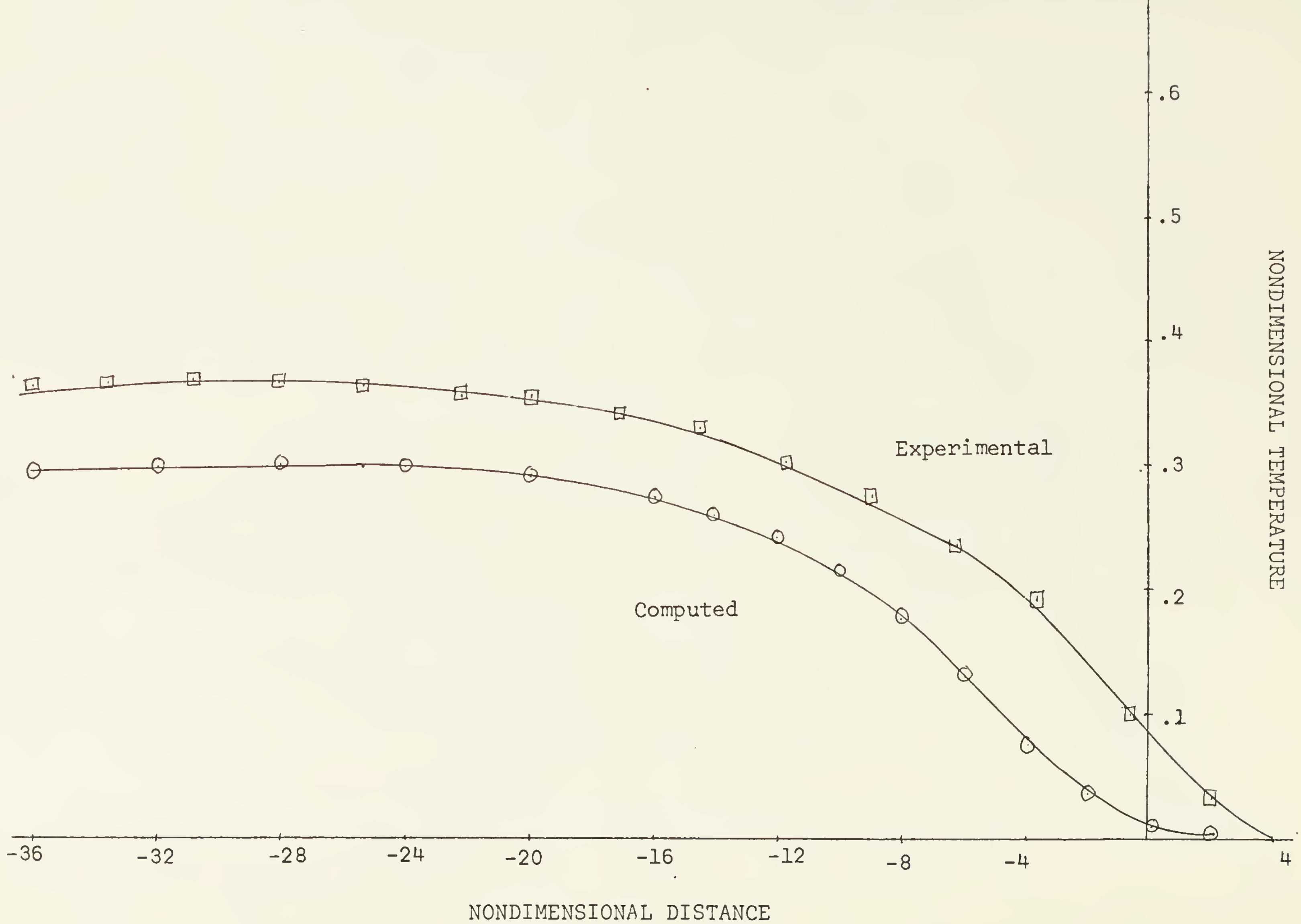


Figure 13
Run #1
Thermocouple 5

Figure 14
Run #1
Thermocouple 6



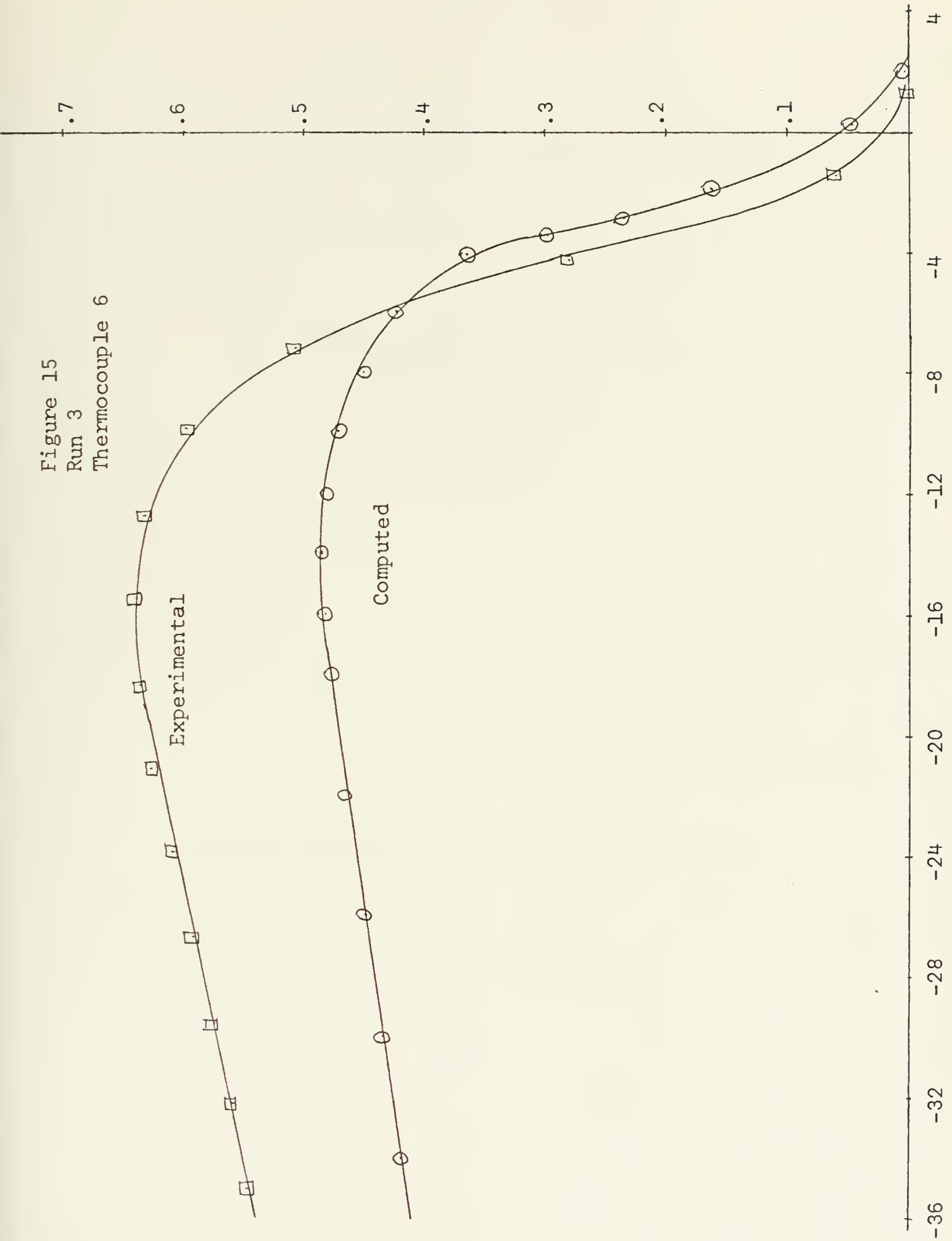
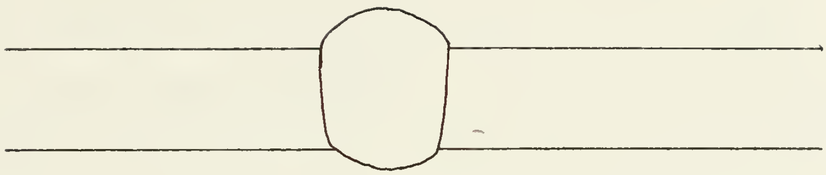
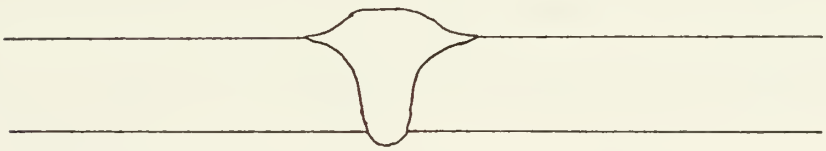


Figure 15
Run 3
Thermocouple 6



TWO-DIMENSIONAL WELD BEAD (SLOW SPEED)



NON-TWO-DIMENSIONAL WELD BEAD (HIGH SPEED)

Figure 16.

VII. SUMMARY AND RECOMMENDATIONS

This method of prediction of temperature history in the heat affected zone is certainly useful because it is accurate. However, because of the constraint of two dimensionality it is, in the opinion of the author, of only limited usefulness. For any sort of temperature prediction process to have any practical usefulness it must predict the temperatures under the actual welding conditions. Actual welding conditions in this case imply a somewhat lesser heat input per unit time than is necessary to achieve two dimensionality. The obvious implication is that this method should be extended into three dimensions. Only speculation is possible, however, whether the practical aspects of this method could be applied in three dimensions. It is doubtful, for example, that the blow-out system would work as well as it did. Since the blow-out equipment is obligated to be angled down toward the weld pool, it might be impossible to determine the depth of the weld pool to the high degree of accuracy required.

The obvious answer to these difficulties is to make the angle between the plate and the blow-out nozzle as small as possible and increase the pressure of the blow-out gas, but only actual attempts would determine if the method is feasible.

Unfortunately also, even the two-dimensional version used for this analysis required 400,000 bytes of memory to analyze an area one inch by three inches. It is obvious that this requirement grows rapidly as either more area is considered or as the third dimension is included. The need to have a large sophisticated computer greatly restricts the practical

value of this analysis since most production facilities like shipyards or aircraft manufacturing plants do not have such an installation. Of course if peripheral devices are available, they could be used to augment in core memory, but only if the program went through a major revision.

With the advantage of hindsight, a far more reasonable approach seems to be to model the molten pool as a shape composed of many point sources. These point sources would be analytically varied in magnitude through the value of the molten metal to give the proper shape and heat input to the metal. This idea is similar to the analytical sources and sinks used to model shapes in the potential flow of fluids. The advantage here, if such an idea is possible, is that the temperature could be computed at any point in the material without an enormous amount of computer memory being required since the temperature at every point is simply a combination of the temperatures resulting from the action of each small source. It is obvious that the above statement assumes linearity which would have to be checked experimentally. However, for practical applications, it is the opinion of the author that a way other than finite difference solution must be found in order to reduce the computer memory requirements.

VIII. BIBLIOGRAPHY

1. Adams, C., "Cooling Rates and Peak Temperatures in Fusion Welding," Welding Journal, 37, 1958, pp. 210s - 215s.

2. Appes, R.L. and D.R. Milner, "Heat Flow in Argon-Arc Welding," British Welding Journal, 2, 1955, pp. 475 - 485.

3. Christensen, N., V. Davies, and K. Gjermundsen, "The Distribution of Temperatures in Arc Welding," British Welding Journal, 12, 1965, pp. 54 - 75.

4. Nippes, E., "The Weld Heat Affected Zone," Welding Journal, January 1959, p. 1s.

5. Pavelic, V., Temperature Histories in Thin Steel Plate Welded With Tungsten Inert Gas, Ph.D. Thesis, U. of Wisconsin, Madison, 1968.

6. Rosenthal, D., "Mathematical Theory of Heat Distribution During Welding and Cutting," Welding Journal, 20, 1941, pp. 220s - 234s.

7. Tanbakuchi, R., Measured Metal Temperatures During Arc Welding, Ph.D. Thesis, U. of Wisconsin, Madison, 1967.

8. Wells, A., "Heat Flow in Welding," Welding Journal, 31, 1952, pp. 263s - 267s.

APPENDIX A.

PROGRAM GMA WELD

```
C
C **** PROGRAM GMA WELD ****
C
      DIMENSION X(10),Y(10),XNORM(10),YNORM(10),XNFOR(15),TCPL(2)
      DIMENSION XNEWB(50),CKOBF(15),AFOF(15),SLOPE(15),XND(50),YNEWB(50)
      DIMENSION CONTF(15),RFOF(15),WHATF(15),XNFOF(15),YNFOF(15),YND(50)
      DIMENSION XNFOB(50),YNFOB(50),AXF(30,75),AYF(30,75),AXB(150,75)
      DIMENSION ONF(30,75),ONB(150,75)
      DIMENSION XGRPL(50),YGRPL(50),XTGR(200),TGRP(200),JGRP(100)
      DIMENSION MAJBOR(50),XPF(50)
      DIMENSION AYB(150,75),TNF(30,75),TNB(150,75),NPRT(3),XPB(150)
      DIMENSION BK1RD(60)
      DIMENSION BIORD(60),BI1RD(60),BKORD(60),RRED(60)
      DIMENSION TRED(50),TKRD(50),TDRD(50)
      DIMENSION TNFA(30,75),TNFB(30,75),TNBA(150,75),TNBB(150,75)
      COMMON BIORD,BI1RD,BKORD,BK1RD,RRED
      COMMON TRED,TKRD,TDRD
      COMMON TMELT,TAMB,ALMIN,TDIFF,DISTN,CCONV,CRAD,VEL,THICK
      COMMON TKXTD,TDXTD,TKMTD,TDMTD,NOPRP
      DRIVE(A,B,C,D)=(A-B)/(C-D)
      CIRCLF(RCIR,ARCIR)=SQRT(RCIR**2-ACIR**2)
      SLINE(SLA,SLARG,SLB,SLC)=SLA*SLARG+SLB*SLC
 2000 FORMAT(I3)
 2850 FORMAT(3F10.5,E12.4)
 3000 FORMAT(3F10.5)
 4000 FORMAT(5F10.5)
 5000 FORMAT(4F10.5)
 6000 FORMAT(2F10.5)
 7000 FORMAT(F10.5)
      READ(5,4000) (RRED(I),BIORD(I),BI1RD(I),BKORD(I),BK1RD(I),I=1,60)
      READ(5,2000) NOPRP
      READ(5,3000) (TRED(I),TKRD(I),TDRD(I),I=1,NOPRP)
      READ(5,7000) SPDCOR
      READ(5,6000) VOLT,AMP
      READ(5,3000) TMELT,THIKRD,SPD
      READ(5,2850) TAMB,TUNN,CCONV,CRAD
```

```
      READ(5,3000) XLONG,YLONG,XLONE
      READ(5,5000) GRX12,GRX34,YMULT1,YMULT2
      READ(5,2000) NWANT
      READ(5,2000) ITCHK
      READ(5,2000) ITERNT
      READ(5,2000)IFPRT
      READ(5,2000) NPONT
      READ(5,2000) NEVIS1
      READ(5,2000) NEVIS2
      READ(5,2000) NEVISI
      READ(5,7000)EPSI
      READ(5,7000) AKHAR
 3889 READ(5,2000)NEXPP
      READ(5,6000) (X(I),Y(I),I=1,NEXPP)
      READ(5,7000) CONVG
      READ(5,2000) NGO
C FIND THE MINIMUM VALUE OF DIFFUSIVITY
      ALMIN=TDRD(1)
      DO 1313 I=2,NOPRP
      IF(ALMIN-TDRD(I)) 1313,1313,1314
 1314 ALMIN=TDRD(I)
 1313 CONTINUE
C NORMALIZING OF ALL INPUT DATA
      TKXTD=DRIVE(TKPD(NOPRP),TKRD(NOPRP-1),TRED(NOPRP),TRED(NOPRP-1))
      TDXTD=DRIVE(TDPD(NOPRP),TDRD(NOPRP-1),TRED(NOPRP),TRED(NOPRP-1))
      TKMTD=DRIVE(TKPD(1),TKRD(2),TRED(1),TRED(2))
      TDMTD=DRIVE(TDPD(1),TDRD(2),TRED(1),TRED(2))
      IGRPL=0
      THICK=THIKPD/12.
      VEL=5.*SPD
      DISTN=SPDCOR*ALMIN/VEL
      DISIN=12.*DISTN
      TDIFF=TMELT-TAMB
      TN=(TUNN-TAMB)/TDIFF
      WRITE(6,8001)
 8001 FORMAT('1',9X,'****** WELDING CONDITIONS*******')
```

```
      WRITE(6,8002) VOLT,AMP,VEL
 8002 FORMAT('OVOLTAGE=',F10.5,' VOLTS   CURRENT=',F10.5,' AMPS   SPEED=
     1',F10.5,' FT./HR.')
      WRITE(6,8003) THIKRD
 8003 FORMAT('0',//,' METAL THICKNESS=',F8.5,' IN.')
      WRITE(6,8004) CCONV,CRAD
 8004 FORMAT('0',//,' CONVECTIVE COEFFICIENT=',F10.5,'   RADIATION COEFF
     1ICIENT=',F12.4)
      WRITE(6,8011)
 8011 FORMAT('0*** PUDDLE SIZE ***',/,'     X(I)',8X,'Y(I)')
      WRITE(6,8012)(X(I),Y(I),I=1,NEXPP)
 8012 FORMAT(' ',F9.5,3X,F10.5)
      WRITE(6,8005) DISIN
 8005 FORMAT('OLENGTH NORMALIZING FACTOR',F10.5,' IN.')
C NORMALIZE THE WELD POOL DIMENSIONS
      DO 1001 I=1,NEXPP
      XNORM(I)=X(I)/DISIN
 1001 YNORM(I)=Y(I)/DISIN
C  FIND THE ORIGIN
      IF(NEXPP-2) 1966,1966,1967
 1967 READ (5,6000) XWRD,YWRD
      XWRD=XWRD/DISIN
      YWRD=YWRD/DISIN
      CALL PRBLW(RF,AO,XNORM(NEXPP),XWRD,YWRD,XWRD/2.,SCALE)
      GO TO 1968
 1966 SCALE=1.
      CALL ORIGIN(RF,ALMIN/ALMIN,XNORM(NEXPP),XNORM(NEXPP)/2.,TN,SCALE)
C SHIFT THE EXPERIMENTAL POINTS OVER TO ORIGIN JUST FOUND
 1968 DO 1002 I=1,NEXPP
 1002 XNFOR(I)=RF-XNORM(I)
C GRID SET UP STARTS HERE
C NTCPL SIGNIFIES POINTS WHERE TEMPERATURE IS DESIRED (MAX OF 2)
      READ(5,2000) NTCPL
      IF(NTCPL-1) 1801,1800,1800
 1801 GRY13=YMULT1
      GRY24=YMULT2
```

```
      YLG13=YLONG/2.
      GO TO 1805
 1800 READ(5,7000) (TCPL(I),I=1,NTCPL)
      READ(5,6000) FACTOR,RHOTSP
      QTOT=FACTOR*VOLT*AMP*3.415
      QC2=432./RHOTSP**2
      QC1=(QC2*QTOT)/3.14159
      WRITE(6,8006)
 8006 FORMAT('OTHERMOCOUPLE LOCATIONS',//' TCPL   IN. FROM WELD C/L')
      WRITE(6,8007)(I,TCPL(I),I=1,NTCPL)
 8007 FORMAT('  ',I2,7X,F8.5)
      IF(NTCPL-1) 1801,1802,1803
 1802 GRY24=YMULT2
      GO TO 1804
 1803 GRY24=AUTSCL(TCPL(2)-TCPL(1),DISIN*YMULT2)/DISIN
 1804 GRY13=AUTSCL(TCPL(1),DISIN*YMULT1)/DISIN
      YLG13=TCPL(1)
 1805 NOINF=(XLONF/DISIN+RF)/GRX12+1.
      NOINR=((XLONG-XLONF)/DISIN-RF)/GRX34+1.
      NOIP1=YLG13/(DISIN*GRY13)+1.
      NOIP2=(YLONG-YLG13)/(DISIN*GRY24)
      NOIPT=NOIP1+NOIP2
      NIP1T=NOIP1-1
      NOVTF=NOINF-1
      NVIPT=NOIPT-1
      NOVTB=NOINR-1
      NOINB=NOINR
      WRITE(6,8009) NOIP1
 8009 FORMAT('OFIRST THERMOCOUPLE IS IN COLUMN',I3)
      IF(NTCPL-1) 1705,1705,1706
 1705 NPRT(3)=NVIPT
      GO TO 1707
 1706 NPRT(3)=(TCPL(2)-TCPL(1))/(DISIN*GRY24)+TCPL(1)/(DISIN*GRY13)+1.
      WRITE(6,8010) NPRT(3)
 8010 FORMAT('OSECOND THERMOCOUPLE IS IN COLUMN',I3)
 1707 NPRT(2)=NOIP1
```

```
      NPRT(1)=1
      WRITE(6,8018)
 8018 FORMAT('0',///,10X,'******* GRID DATA *******',//,' DIRECTION
     1  NO. OF NODES   SPACING FACTOR')
      WRITE(6,8014) NOVTF,GRX12
      WRITE (6,8015) NOVTB,GRX34
      WRITE(6,8016) NVIPT,GRY13
      WRITE(6,8017) NVIPT,GRY24
 8014 FORMAT(' ',4X,'X',11X,I3,11X,F10.5,6X,'IN FRONT OF THE ARC')
 8015 FORMAT(' ',4X,'X',11X,I3,11X,F10.5,6X,'BEHIND THE ARC')
 8016 FORMAT(' ',4X,'Y',11X,I3,11X,F10.5,6X,'INSIDE T/C 1')
 8017 FORMAT(' ',4X,'Y',11X,I3,11X,F10.5,6X,'OUTSIDE T/C 1')
      WRITE(6,8008) FACTOR,RHOTSP
 8008 FORMAT('OASSUMED WELDING EFFICIENCY',F8.3,//,' RADIUS OF LUMINOSIT
     1Y',F8.5,' IN.')
      DO 1003 J=1,NIP1T
      DO 1003 I=1,NOVTF
      AXF(I,J)=GRX12
      AYF(I,J)=GRY13
      TNF(I,J)=0.0
      ONF(I,J)=0.0
 1003 CONTINUE
      DO 1004 J=NOIP1,NVIPT
      DO 1004 I=1,NOVTF
      AXF(I,J)=GRX12
      AYF(I,J)=GRY24
      TNF(I,J)=0.0
      ONF(I,J)=0.0
 1004 CONTINUE
      DO 1005 J=1,NIP1T
      DO 1005 I=1,NOVTB
      AXB(I,J)=GRX34
      AYB(I,J)=GRY13
      ONB(I,J)=0.0
      TNB(I,J)=0.0
 1005 CONTINUE
```

```
      DO 1006 J=NOIP1,NVIPT
      DO 1006 I=1,NOVTB
      AXB(I,J)=GRX34
      AYB(I,J)=GRY24
      ONB(I,J)=0.0
      TNB(I,J)=0.0
 1006 CONTINUE
      CHECK=0.
      DO 1007 I=1,NEXPP
      IF(XNEOR(I)) 1008,1008,1007
 1008 IF(CHECK-1.) 1009,1007,1007
 1009 CHECK=CHECK+1.
      NOPIF=I-1
 1007 CONTINUE
C     FIND THE INTERSECTION OF THE POOL WITH THE Y-AXIS
      CALL TYPFN(XNEOR(NOPIF),YNORM(NOPIF),XNEOR(NOPIF+1),YNORM(NOPIF+1)
     1,TN,ACENT,CSKOC,SLOPC,CONSC,RCEN,WHATC,REGIN,R1,R2,SCALE)
C THIS SUBROUTINE DETERMINES WHAT TYPE OF FUNCTION SHOULD BE USED TO FIT
C THE GIVEN DATA POINTS
      IF(WHATC-1.) 1030,1031,1030
 1030 IF(WHATC-2.) 1032,1033,1032
 1031 CALL COLUMN(YPS,0.,ACENT,CSKOC,1.,TN,SCALE)
      IF(NEXPP-2) 1034,1034,9117
 9117 IF(ABS(YPS)-YNORM(NOPIF+1)) 1034,1034,1032
 1033 YPS=RCEN
      GO TO 1034
 1032 YPS=SLINE(SLOPC,0.,1.,CONSC)
 1034 JATOF=YPS/GRY13+1.
      ZAROR=JATOF
      AYF(1,JATOF)=ZAROR*GRY13-YPS
      AYB(1,JATOF)=AYF(1,JATOF)
      IGRPL=IGRPL+1
      XGRPL(IGRPL)=0.
      YGRPL(IGRPL)=YPS
      DO 1100 J=1,JATOF
      TNF(1,J)=TN
```

```
 1100 TNB(1,J)=TN
      IF(JATOF-1) 1021,1021,1022
 1022 AYF(1,JATOF-1)=2.*GRY13-AYF(1,JATOF)
      AYB(1,JATOF-1)=AYF(1,JATOF-1)
 1021 NTPCF=NOPIF+1
C START OF NUMBERING OF EXP POINTS AT ORIGIN
      DO 1010 I=1,NTPCF
      IF(I-1) 1071,1071,1072
 1071 XNFOF(I)=0.
      YNFOF(I)=YPS
      GO TO 1010
 1072 NVTF=NTPCF-I+1
      XNFOF(I)=XNFOP(NVTF)
      YNFOF(I)=YNORM(NVTF)
 1010 CONTINUE
C START OF FITTING OF CURVE THROUGH FORWARD POINTS
      NOF1=NOPIF
      NOF2=NOF1
 2008 NEXTM=0
      DO 2001 I=1,NOF1
      IF(NEXTM-1) 2003,2002,2002
 2003 CALL PNTEND(XNFOF(I),YNFOF(I),XNFOF(I+1),YNFOF(I+1),GRX12,GRY13/2.
     1,TN,NWANT,NEXSE,XBT,YBT,SCALE)
      IF(NEXSE-1) 2001,2004,2004
 2004 NEXTM=NEXSE
      I2=I+1
      NOF2=NOF1+NEXSE
      XND(I)=XNFOF(I)
      YND(I)=YNFOF(I)
      XND(I2)=XBT
      YND(I2)=YBT
      DO 2005 L=I2,NOF2
      XND(L+1)=XNFOF(L)
 2005 YND(L+1)=YNFOF(L)
      NOF3=NOF2+1
      DO 2009 LL=I,NOF3
```

```
      XNFOF(LL)=XND(LL)
 2009 YNFOF(LL)=YND(LL)
 2001 CONTINUE
 2002 IF(NOF2-NOF1) 2006,2006,2007
 2007 NOF1=NOF2
      GO TO 2008
 2006 NTPCF=NOF1+1
      NOPIF=NOF1
      ZARCL=0.
      DO 1011 I=1,NTPCF
      IF(NOPIF-I) 1011,1070,1070
 1070 CALL TYPEN(XNFOF(I),YNFOF(I),XNFOF(I+1),YNFOF(I+1),TN,AFOF(I),
     1CKOBF(I),SLOPE(I),CONTF(I),RFOF(I),WHATF(I),REGE,RA,RB,SCALE)
      NCOLA=XNFOF(I)/GRX12
      NCOLB=XNFOF(I+1)/GRX12
      FDLCN=NCOLA-NCOLB
      NCLDF=ABS(FDLCN)
      NCNEW=NCLDF+1
      DO 1012 K=1,NCNEW
      IF(NCLDF-K) 1012,1073,1073
 1073 ZARCL=ZARCL+1.
      IGRPL=IGRPL+1
      IF(ABS(XNFOF(I+1)-ZARCL*GRX12)-GRX12/20.) 1401,1401,1402
 1402 IF(WHATF(I)-1.) 1035,1036,1035
 1035 IF(WHATF(I)-2.) 1037,1038,1037
 1036 CALL COLUMN(YCALF,ZARCL*GRX12,AFOF(I),CKOBF(I),(YNFOF(I+1)+YNFOF(I
     1))/2.,TN,SCALE)
      IF(YCALF-(YNFOF(I)+GRY13)) 1039,1039,1037
 1038 YCALF=CIRCLE(RFOF(I),ZARCL*GRX12)
      GO TO 1039
 1037 YCALF=SLINE(SLOPE(I),ZARCL*GRX12,1.,CONTF(I))
      GO TO 1039
 1401 YCALF=YNFOF(I+1)
 1039 JCALF=YCALF/GRY13+1.
      ICALF=ZARCL+1.
      ZARIF=JCALF
```

```
      AYF(ICALF,JCALF)=ZARIF*GRY13-YCALF
      XGRPL(IGRPL)=ZARCL*GRX12
      YGRPL(IGRPL)=YCALF
      DO 1101 J=1,JCALF
 1101 TNF(ICALF,J)=TN
      IF(JCALF-1) 1012,1012,1024
 1024 AYF(ICALF,JCALF-1)=2.*GRY13-AYF(ICALF,JCALF)
 1012 CONTINUE
 1011 CONTINUE
      ICENF=XNFOF(NTPCF)/GRX12+1.
      ZARCF=ICENF
      AXF(ICENF,1)=ZARCF*GRX12-XNFOF(NTPCF)
      TNF(ICENF,1)=TN
      IF(ICENF-1) 1025,1025,1026
 1026 AXF(ICENF-1,1)=2.*GRX12-AXF(ICENF,1)
 1025 ZARRW=0.
      DO 1013 I=1,NTPCF
      IF(NDRIF-I) 1013,1080,1080
 1080 NK=NTPCF-I+1
      NROWA=YNFOF(NK)/GRY13
      NROWB=YNFOF(NK-1)/GRY13
      FDORN=NROWA-NROWB
      NRODE=ABS(FDORN)
      NRNEW=NRODE+1
      DO 1014 KR=1,NRNEW
      IF(NRODE-KR) 1014,1081,1081
 1081 ZARRW=ZARRW+1.
      IGRPL=IGRPL+1
      IF(ABS(YNFOF(NK-1)-ZARRW*GRY13)-GRY13/20.) 1403,1403,1404
 1404 IF(WHATF(NK-1)-1.) 1040,1041,1040
 1040 IF(WHATF(NK-1)-2.) 1042,1043,1042
 1041 CALL POWPS(XCALF,ZARRW*GRY13,AFOF(NK-1),CKOBF(NK-1),(XNFOF(NK)+XNF
     1OF(NK-1))/2.,TN,SCALE)
      IF(XCALF-(XNFOF(NK)+GRX12)) 1044,1044,1042
 1043 XCALF=CIRCLE(RFOF(NK-1),ZARRW*GRY13)
      GO TO 1044
```

```
 1042 XCALF=SLINE(1./SLOPE(NK-1),ZARRW*GRY13,-1./SLOPE(NK-1),CONTF(NK-1)
     1)
      GO TO 1044
 1403 XCALF=XNFOF(NK-1)
 1044 ICALR=XCALF/GRX12+1.
      JCALR=ZARRW+1.
      ZARIR=ICALR
      AXF(ICALR,JCALR)=ZARIR*GRX12-XCALF
      XGRPL(IGRPL)=XCALF
      YGRPL(IGRPL)=ZARRW*GRY13
      IF(ICALR-1) 1014,1014,1027
 1027 AXF(ICALR-1,JCALR)=2.*GRX12-AXF(ICALR,JCALR)
 1014 CONTINUE
 1013 CONTINUE
      WRITE(6,1250)
 1250 FORMAT('1*** NONDIMENSIONAL WELD PUDDLE BOUNDARY***',///)
      WRITE (6,1251)
 1251 FORMAT('     X(I)',8X,'Y(I)')
      WRITE(6,1252) (XNFOF(I),YNFOF(I),I=1,ICENF)
 1252 FORMAT(' ',2F12.5)
C CURVE FITTING IN FRONT OF THE ARC IC COMPLETE
      XNFOB(1)=0.
      YNFOB(1)=YPS
      NOPIB=NEXPP-NOPIF
      NTPCB=NOPIB+1
      NOMRE=0.
      DO 1015 I=2,NTPCB
      NOMRE=NOMRE+1
      NVTB=NOPIF+NOMRE
      XNFOB(I)=XNFOR(NVTB)
 1015 YNFOB(I)=YNORM(NVTB)
      NOBA=NOPIB
      NOBB=NOBA
 3008 NEXTM=0
      DO 3001 I=1,NOBA
      IF(NEXTM-1) 3003,3002,3002
```

```
3003 CALL PNTEND(XNFOB(I),YNFOB(I),XNFOB(I+1),YNFOB(I+1),GRX34,GRY13/2.
    1,TN,NWANT,NEXB,XBTB,YBTB,SCALE)
3010 IF(NEXB-1) 3001,3004,3004
3004 NEXTM=1
     IB=I+1
     NOBB=NOBA+1
     XND(I)=XNFOB(I)
     YND(I)=YNFOB(I)
     XND(IB)=XBTB
     YND(IB)=YBTB
     DO 3005 IC=IB,NOBB
     XND(IC+1)=XNFOB(IC)
3005 YND(IC+1)=YNFOB(IC)
     NOBC=NOBB+1
     DO 3009 ID=I,NOBC
     XNFOB(ID)=XND(ID)
3009 YNFOB(ID)=YND(ID)
3001 CONTINUE
3002 IF(NOBB-NOBA) 3006,3006,3007
3007 NOBA=NOBB
     GO TO 3008
3006 NTPCB=NOBA+1
     NOPIB=NOBA
     ZARCB=0.
     JMAX=1
     DO 1016 I=1,NTPCB
     IF(NOPIB-I) 1016,1090,1090
1090 CALL TYPEN(XNFOB(I),YNFOB(I),XNFOB(I+1),YNFOB(I+1),TN,AFCB,CKOBB,
    1SLOPB,CONTB,REFB,WHATB,REGB,RA,RB,SCALE)
     NCOLA=XNFOB(I)/GRX34
     NCOLB=XNFOB(I+1)/GRX34
     FDLCN=NCOLA-NCOLB
     NCLDF=ABS(FDLCN)
     NBNEW=NCLDF+1
     DO 1017 K=1,NBNEW
     IF(NCLDF-K) 1017,1091,1091
```

```
1091 ZARCB=ZARCB-1.
     IGRPL=IGRPL+1
     IF(ABS(XNFOB(I+1)-ZARCB*GRX34)-GRX34/20.) 1405,1405,1406
1406 IF(WHATB-1.) 1045,1046,1045
1045 IF(WHATB-2.) 1047,1048,1047
1046 CALL COLUMN(YCALB,ZARCB*GRX34,AFOB,CKOBB,(YNFOB(I+1)+YNFOB(I))/2.,
    1TN,SCALE)
     IF(YCALB-(YNFOB(I)+2.*GRY13)) 1049,1049,1047
1048 YCALB=CIRCLE(RFOB,ZARCB*GRX34)
     GO TO 1049
1047 YCALB=SLINE(SLOPB,ZARCB*GRX34,1.,CONTB)
     GO TO 1049
1405 YCALB=YNFOB(I+1)
1049 JCALB=YCALB/GRY13+1.
     IF(JMAX-JCALB) 1096,1097,1097
1096 JMAX=JCALB
     IMAX=-ZARCB+1.
1097 ICALB=-ZARCB+1.
     ZARIB=JCALB
     AYB(ICALB,JCALB)=ZARIB*GRY13-YCALB
     XGRPL(IGRPL)=ZARCB*GRX34
     YGRPL(IGRPL)=YCALB
     DO 1102 J=1,JCALB
1102 TNB(ICALB,J)=TN
     XNEWB(ICALB)=ZARCB*GRX34
     YNEWB(ICALB)=YCALB
     IF(JCALB-1) 1017,1017,1028
1028 AYB(ICALB,JCALB-1)=2.*GRY13-AYB(ICALB,JCALB)
1017 CONTINUE
1016 CONTINUE
     ICENB=ABS(XNFOB(NTPCB))/GRX34+1.
     ZARCB=ICENB
     AXB(ICENB,1)=ZARCB*GRX34-ABS(XNFOB(NTPCB))
     TNB(ICENB,1)=TN
     IF(ICENB-1) 1060,1060,1061
1061 AXB(ICENB-1,1)=2.*GRX34-AXB(ICENB,1)
```

```
1060 ZARWB=0.
     IF(AXB(ICENB,1)-GRX34) 1501,1502,1501
1502 ICNEW=ICENB
     GO TO 1503
1501 ICNEW=ICENB+1
1503 XNEWB(ICNEW)=XNFOB(NTPCB)
     YNEWB(ICNEW)=0.
     XNEWB(1)=0.
     YNEWB(1)=YPS
     APW=1.
     OVER=0.
     DO 1018 I=2,ICNEW
1092 NKB=ICNEW-I+2
     NRWBA=YNEWB(NKB)/GRY13
     NRWBB=YNEWB(NKB-1)/GRY13
     BDORN=NRWBA-NRWBB
     NRODB=ABS(BDORN)
     IF(NRODB-1) 1018,1093,1093
1093 CALL TYPEN(XNEWB(NKB),YNEWB(NKB),XNEWB(NKB-1),YNEWB(NKB-1),TN,AFOB
    1,CKOBB,SLOPB,CONTB,RFOB,WHATB,REGB,RA,RB,SCALE)
     DO 1019 KR=1,NRODB
     ZARWB=ZARWB+APW
     IGRPL=IGRPL+1
     IF(ABS(YNEWB(NKB-1)-ZARWB*GRY13)-GRY13/20.) 1407,1407,1408
1408 IF(WHATB-1.) 1050,1051,1050
1050 IF(WHATB-2.) 1052,1053,1052
1051 CALL POWPS(XCALB,ZARWB*GRY13,AFOB,CKOBB,(XNEWB(NKB)+XNEWB(NKB-1))/
    12.,TN,SCALE)
     IF(ABS(XCALB)-ABS(XNEWB(NKB)-GRX34)) 1054,1054,1052
1053 XCALB=CIRCLE(RFOB,ZARWB*GRY13)
     GO TO 1054
1052 XCALB=SLINE(1./SLOPB,ZARWB*GRY13,-1./SLOPB,CONTB)
     GO TO 1054
1407 XCALB=YNEWB(NKB-1)
1054 ICALB=ABS(XCALB/GRX34)+1.
     ZARIB=ICALB
```

```
      JCALB=ZARWB+1.
      IF(JCALB-JMAX) 1098,1089,1099
 1089 IF(OVER-1.) 1088,1086,1086
 1088 ARW=0.
      OVER=1.
      GO TO 1087
 1098 IF(OVER-1.) 1085,1086,1086
 1099 OVER=1.
 1086 ARW=-1.
      GO TO 1087
 1085 ARW=1.
 1087 AXB(ICALB,JCALB)=ZARIB*GRX34-ABS(XCALB)
      XGRPL(IGRPL)=XCALB
      YGRPL(IGRPL)=ZARWB*GRY13
      IF(ICALB-1) 1019,1019,1029
 1029 AXB(ICALB-1,JCALB)=2.*GRX34-AXB(ICALB,JCALB)
 1019 CONTINUE
 1018 CONTINUE
 1604 IMAXA=IMAX-1
      DO 1601 J=1,JMAX
      DO 1683 I=1,IMAX
      IF(TNB(I,J)-TN) 1682,1601,1682
 1682 IF(AXB(I,J)-GRX34) 1686,1683,1685
 1686 AXB(I,J)=GRX34-AXB(I,J)
      GO TO 1601
 1685 AXB(I,J)=GRX34
 1683 CONTINUE
 1601 CONTINUE
      WRITE(6,1252) (XNEWB(I),YNEWB(I),I=1,ICNEW)
C GRID SET ENDS HERE
      DO 3669 IB=1,NOINF
      ZBGR=NOINF-IB
 3669 XTGR(IB)=GRX12*ZBGR
      DO 3679 IB=1,NOVTB
      ZBGR=IB
      ID=NOINF+IB
```

```
 3679 XTGR(ID)=-GRX34*ZRGR
      IGT=NOINB+NOVTF
      MEG=IGT-NOPNT
C START INITILIZATION OF POINTS IN FRONT OF THE ARC
      WRITE(6,8019)
 8019 FORMAT('OSTART')
      CALL BESSEL(-XNFOR(NEXPP),BIOJ,BIIJ,BKOJ,BKIJ)
      IF(BKOJ.LT..1E-50) BKOJ=.1E-50
      ATMP=1.
      CTMP=TN*EXP(ATMP*XNFOR(NEXPP))/(AKHAR*BKOJ)
      DO 2101 J=1,NVIPT
      AXB(NOVTB,J)=.1*GRX34
      TNF(NOINF,J)=0.
 2101 TNB(NOINB,J)=0.
      DO 2102 I=1,NOINF
      AYF(I,NVIPT)=.1*GRY24
 2102 TNF(I,NOIPT)=0.
      DO 2103 I=1,NOINB
      AYB(I,NVIPT)=.1*GRY24
 2103 TNB(I,NOIPT)=0.
      DO 1200 J=1,NOIP1
      DO 1200 I=1,NOINF
      IF(TNF(I,J)-TN) 1201,1200,1209
 1201 ZARX=I-1
      ZARY=J-1
      TNF(I,J)=DRAJH(CTMP,ATMP,ZARX*GRX12,ZARY*GRY13)
      QNF(I,J)=QC1*EXP((-QC2*DISTN**2)*((ZARX*GRX12)**2+(ZARY*GRY13)**2
     1))
      IF(TNF(I,J)-TN) 1200,1209,1209
 1209 TNF(I,J)=TN-1.
 1200 CONTINUE
      ZANPA=NOIP1-1
      NODPB=NOIP2+1
      DO 1202 K=2,NODPB
      DO 1202 I=1,NOINF
      J=K+NOIP1-1
```

```
      IF(TNF(I,J)-TN) 1203,1202,1210
 1203 ZARX=I-1
      ZARY=K-1
      TNF(I,J)=DRAJH(CTMP,ATMP,ZARX*GRX12,ZANPA*GRY13+ZARY*GRY24)
      QNF(I,J)=QC1*EXP((-QC2*DISTN**2)*((ZARX*GRX12)**2+(ZANPA*GRY13+
     1ZARY*GRY24)**2))
      IF(TNF(I,J)-TN) 1202,1210,1210
 1210 TNF(I,J)=TN-.1
 1202 CONTINUE
C START INITILIZING POINTS IN BACK OF THE ARC
      DO 1204 J=1,NOIP1
      DO 1204 I=1,NOINB
      IF(TNB(I,J)-TN) 1205,1204,1211
 1205 ZARX=1-I
      ZARY=J-1
      TNB(I,J)=DRAJH(CTMP,ATMP,ZARX*GRX34,ZARY*GRY13)
      QNB(I,J)=QC1*EXP((-QC2*DISTN**2)*((ZARX*GRX34)**2+(ZARY*GRY13)**2
     1))
      IF(TNB(I,J)-TN) 1204,1211,1211
 1211 TNB(I,J)=TN-.1
 1204 CONTINUE
      DO 1206 K=2,NODPB
      DO 1206 I=1,NOINB
      J=K+NOIP1-1
      IF(TNB(I,J)-TN) 1207,1206,1212
 1207 ZARX=1-I
      ZARY=K-1
      TNB(I,J)=DRAJH(CTMP,ATMP,ZARX*GRX34,ZANPA*GRY13+ZARY*GRY24)
      QNB(I,J)=QC1*EXP((-QC2*DISTN**2)*((ZARX*GRX34)**2+(ZANPA*GRY13+
     1ZARY*GRY24)**2))
      IF(TNB(I,J)-TN) 1206,1212,1212
 1212 TNB(I,J)=TN-.1
 1206 CONTINUE
C TEMPERATURE DISTRIBUTION BASED ON POINT SOURCE THEORY
      DO 4055 I=1,NOINB
      ZARIB=I-1
```

```
4055 XPB(I)=-ZARIB*GRX34
     DO 4056 I=1,NOINF
     ZARIB=I-1
4056 XPF(I)=ZARIB*GRX12
     DO 4051 J=NEVIS1,NEVIS2,NEVISI
     IF(J-NOIP1) 4052,4052,4053
4052 ZARIB=J-1
     YPRNT=ZARIB*GRY13
     IF(NGO.GT.0) GO TO 1328
     GO TO 4054
4053 ZARIB=NOIP1-1
     ZARBB=J-NOIP1
     YPRNT=ZARIB*GRY13+ZARBB*GRY24
4050 FORMAT('1',9X,'******* TEMPERATURE DISTRIBUTION BASED ON POINT SOU
    1RCE THEORY *******',///)
4054 WRITE(6,4050)
     WRITE(6,3998)
3998 FORMAT(' COLUMN',3X,'COR. D-L DIS.',3X,'ROW',3X,'COR. D-L DIS.',6X
    1,'X-IN.',6X,'Y-IN.',6X,'TEMP')
3999 FORMAT(' ',I4,6X,F8.3,6X,I3,5X,F8.3,6X,F10.5,3X,F10.5,3X,F10.5)
     WRITE(6,3999) (J,YPRNT,I,XPF(I),AXF(I,J),AYF(I,J),TNF(I,J),I=1,NOV
    1TF)
     WRITE(6,3999) (J,YPRNT,I,XPB(I),AXB(I,J),AYB(I,J),TNB(I,J),I=1,NOV
    1TB)
4051 CONTINUE
1328 ITER=-1
     DO 9011 J=1,NVIPT
9011 MAJBOR(J)=IMAX
1701 IF(ITER-ITCHK) 1704,1702,1703
1704 ITER=ITER+1
     DO 1307 J=1,NVIPT
     IJDID=MAJBOR(J)
     DO 1300 I=IMAX,IJDID
     IF(TNB(I,J)-TN) 1304,1300,1304
1304 IF(J-1) 1300,1301,1302
1301 BBMN=AYB(I,J)
```

```
      TDMN=TNB(I,J+1)
      GO TO 1303
 1302 BBMN=AYB(I,J-1)
      TDMN=TNB(I,J-1)
 1303 CALL DIFEQ(TNB(I,J),TNB(I,J),TNB(I-1,J),TNB(I+1,J),TNB(I,J+1),
     1TDMN,AXB(I-1,J),AXB(I,J),AYB(I,J),BBMN,TN,QNB(I,J))
 1300 CONTINUE
      DO 9009 I=IJDID,NOVTB
      IF(TNB(I,J)-TN) 9004,9009,9004
 9004 IF(J-1) 9009,9001,9002
 9001 BBMN=AYB(I,J)
      TDMN=TNB(I,J+1)
      GO TO 9003
 9002 BBMN=AYB(I,J-1)
      TDMN=TNB(I,J-1)
 9003 CALL DIFEQ(TNBDI,TNB(I,J),TNB(I-1,J),TNB(I+1,J),TNB(I,J+1),TDMN,
     1AXB(I-1,J),AXB(I,J),AYB(I,J),BBMN,TN,QNB(I,J))
      IF(ABS(TNBDI-TNB(I,J))-EPSI) 9005,9005,9006
 9006 TNB(I,J)=TNBDI
 9009 CONTINUE
 9005 IF(I-NOVTB) 9013,9023,9023
 9023 MAJBOR(J)=I
      TNB(NDINB,J)=1.1*TNB(NOVTB,J)-.1*TNB(NOVTB-1,J)
      GO TO 9033
 9013 MAJBOR(J)=I+1
 9033 IF(IMAX-2) 1605,1605,1606
 1606 DO 1305 I1=2,IMAXA
      I=IMAX-I1+1
      IF(TNB(I,J)-TN) 1306,1305,1306
 1306 CALL DIFEQ(TNB(I,J),TNB(I,J),TNB(I-1,J),TNB(I+1,J),TNB(I,J+1)
     1,TNB(I,J-1),AXB(I-1,J),AXB(I,J),AYB(I,J),AYB(I,J-1),TN,QNF(I,J))
 1305 CONTINUE
 1605 IF(TNF(1,J)-TN) 2105,2106,2105
 2105 CALL DIFEQ(TNF(1,J),TNF(1,J),TNF(2,J),TNB(2,J),TNF(1,J+1),TNF(1,J-
     11),AXF(1,J),AXB(1,J),AYF(1,J),AYF(1,J-1),TN,QNF(I,J))
      TNB(1,J)=TNF(1,J)
```

```
2106 IF(ITER-ITERPNT) 9043,9043,1307
9043 DO 1312 I=2,NOVTF
     IF(TNF(I,J)-TN) 1308,1312,1308
1308 IF(J-1) 1312,1309,1310
1309 BBMN=AYF(I,J)
     TDMN=TNF(I,J+1)
     GO TO 1311
1310 BBMN=AYF(I,J-1)
     TDMN=TNF(I,J-1)
1311 CALL DIFEQ(TNF(I,J),TNF(I,J),TNF(I+1,J),TNF(I-1,J),TNF(I,J+1),TDMN
    1,AXF(I,J),AXF(I-1,J),AYF(I,J),BBMN,TN,QNF(I,J))
1312 CONTINUE
1307 CONTINUE
     DO 4136 KD=1,NOINB
4136 TNB(KD,NOIPT)=1.1*TNB(KD,NVIPT)-.1*TNB(KD,NVIPT-1)
     DO 4137 KD=2,NOINF
4137 TNF(KD,NOIPT)=1.1*TNF(KD,NVIPT)-TNF(KD,NVIPT-1)*.1
     DO 1326 J=1,NVIPT
     DO 1327 I=1,NOVTB
1327 TNBA(I,J)=TNB(I,J)
1326 CONTINUE
     DO 1315 J=1,NVIPT
     DO 1316 I=1,NOVTF
1316 TNFA(I,J)=TNF(I,J)
1315 CONTINUE
     IF(ITER.EQ.0) GO TO 1321
     DO 1322 J=1,NVIPT
     DO 1323 I=1,NOVTB
     YZ=ABS(TNBA(I,J)-TNBB(I,J))
     IF(YZ.GT.CONVG) GO TO 1321
1323 CONTINUE
1322 CONTINUE
     DO 1324 J=1,NVIPT
     DO 1325 I=1,NOVTF
     XZ=ABS(TNFA(I,J)-TNFB(I,J))
     IF(XZ.GT.CONVG) GO TO 1321
```

```
1325 CONTINUE
1324 CONTINUE
     GO TO 1702
1321 DO 1317 J=1,NVIPT
     DO 1318 I=1,NOVTB
1318 TNBB(I,J)=TNBA(I,J)
1317 CONTINUE
     DO 1319 J=1,NVIPT
     DO 1320 I=1,NOVTF
1320 TNFB(I,J)=TNFA(I,J)
1319 CONTINUE
     GO TO 1701
1702 WRITE(6,8013) ITER
8013 FORMAT('ONUMBER OF ITERATIONS REQUIRED',I5)
     DO 8051 J=NEVIS1,NEVIS2,NEVISI
     IF(J-NOIP1) 8052,8052,8053
8052 ZARIB=J-1
     YPRNT=ZARIB*GRY13
     GO TO 8054
8053 ZARIB=NOIP1-1
     ZARBB=J-NOIP1
     YPRNT=ZARIB*GRY13+ZARBB*GRY24
8054 WRITE(6,8050)
8050 FORMAT('1',9X,'******* TEMPERATURE DISTRIBUTION*******',///)
     WRITE(6,3998)
     WRITE(6,3999) (J,YPRNT,I,XPF(I),AXF(I,J),AYF(I,J),TNF(I,J),I=1,NOV
    1TF)
     WRITE(6,3999) (J,YPRNT,I,XPB(I),AXB(I,J),AYB(I,J),TNB(I,J),I=1,NOV
    1TB)
8051 CONTINUE
1703 STOP
     END
     SUBROUTINE DIFFQ(TNEW,TMP,TAD,TBD,TCD,TDD,AFD,ABD,BFD,BBD,TN,QDIFF
    1)
     DIMENSION BIORD(60),BI1RD(60),BKORD(60),BK1RD(60),RRED(60)
     DIMENSION TRED(50),TKRD(50),TDRD(50)
```

```
      COMMON BIORD,BI1RD,BKORD,BK1RD,RRED
      COMMON TRED,TKPD,TDRD
      COMMON TMELT,TAMB,ALMIN,TDIFF,DISTN,CCONV,CRAD,VEL,THICK
      COMMON TKXTD,TDXTD,TKMTD,TDMTD,NOPRP
      T(A,AA,AB,AF,BA,BF,BB,HLOSS,TKDEP,DIFFR,TA,TB,TC,TD,QDIFU)=(AA*(TA
     1*(1./AF+DIFFR)+TB*(1./BA-DIFFR))+AB*(TC/BF+TD/BB)+TKDEP*(AA**2*(TA
     2-TB)**2+AB**2*(TC-TD)**2)-HLOSS)/A+((DISTN**2)*QDIFU)/(TK*THICK*TD
     3IFF*A)
      CALL PROP(TK,TKDRV,TD,TMP,TRED(1),TRED(1),TRED(NOPRP),TKRD(1),
     1TDRD(1),TKRD(NOPRP),TDRD(NOPRP),TKXTD,TDXTD,TKMTD,TDMTD)
      CALL COEF(AD,AAD,ADD,HLOSD,TKDPD,DIFRD,TMP,AFD,ABD,BFD,BBD,TK,TD,
     1TKDRV)
      TNEW=T(AD,AAD,ADD,AFD,ABD,BFD,BBD,HLOSD,TKDPD,DIFPD,TAD,TBD,TCD,
     1TDD,QDIFF)
      IF(TNEW-1.) 333,332,332
  332 TNEW=.999
  333 RETURN
      END
      SUBROUTINE COEF(A,A1,A2,HLOSS,TKDEP,DIFFP,T,AF,AB,BF,RR,TK,TD,
     1TKDRV)
      DIMENSION BIORD(60),BI1RD(60),BKORD(60),BK1RD(60),RRED(60)
      DIMENSION TRED(50),TKRD(50),TDRD(50)
      COMMON BIORD,BI1RD,BKORD,BK1RD,RRED
      COMMON TRED,TKPD,TDRD
      COMMON TMELT,TAMB,ALMIN,TDIFF,DISTN,CCONV,CRAD,VEL,THICK
      COMMON TKXTD,TDXTD,TKMTD,TDMTD,NOPRP
      A=1./(AF*AB)+1./(BF*RR)
      A1=1./(AF+AB)
      A2=1./(BF+RR)
      HLOSS=DISTN**2/(TK*THICK)*(CCONV*T+CRAD/TDIFF*((T*TDIFF+TAMB)**4-
     1TAMB**4))
      TKDEP=TDIFF*TKDRV/(2.*TK)
      DIFFP=ALMIN/TD
      RETURN
      END
      SUBROUTINE PROP(TK,TKDRV,TD,TND,TRDD,TMIN,TMAX,TKMNT,TDMNT,TKMXT,
```

```
     1TDMXT,TKXDR,TDXDR,TKMDR,TDMDR)
      DIMENSION BIORD(60),BI1RD(60),BKORD(60),BK1RD(60),RRED(60)
      DIMENSION TRED(50),TKRD(50),TDRD(50)
      COMMON BIORD,BI1RD,BKORD,BK1RD,RRED
      COMMON TRED,TKRD,TDRD
      COMMON TMELT,TAMB,ALMIN,TDIFF,DISTN,CCONV,CRAD,VEL,THICK
      COMMON TKXTD,TDXTD,TKMTD,TDMTD,NOPRP
      VINTF(A,B,C,D,F)=A+(B-A)/(C-D)*(F-D)
      VINTD(A,B,C,D)=(A-B)/(C-D)
      T=TND*TDIFF+TAMB
      IF(T-TMIN) 208,208,209
209   IF(T-TMAX) 201,202,202
201   T2=T/TRDD
      I=T2
      IF(TRED(I)/TRDD-T2) 203,204,205
203   IBASE=I
      GO TO 206
205   IBASE=I-1
206   TK=VINTF(TKRD(IBASE),TKRD(IBASE+1),TRED(IBASE+1),TRED(IBASE),T)
      TD=VINTF(TDRD(IBASE),TDRD(IBASE+1),TRED(IBASE+1),TRED(IBASE),T)
      TKDRV=VINTD(TKRD(IBASE),TKRD(IBASE+1),TRED(IBASE),TRED(IBASE+1))
      IF(TD-ALMIN) 210,207,207
204   TK=TKRD(I)
      TD=TDRD(I)
      TKDRV=VINTD(TKRD(I+1),TKRD(I-1),TRED(I+1),TRED(I-1))
      IF(TD-ALMIN) 210,207,207
208   TK=TKMNT+TKMDR*(T-TMIN)
      TD=TDMNT+TDMDR*(T-TMIN)
      TKDRV=TKMDR
      IF(TD-ALMIN) 210,207,207
202   TK=TKMXT+TKXDR*(T-TMAX)
      TD=TDMXT+TDXDR*(T-TMAX)
      TKDRV=TKXDR
      IF(TD-ALMIN) 210,207,207
210   TD=ALMIN
207   RETURN
```

```
      END
      SUBROUTINE TYPEN(X1,Y1IN,X2,Y2IN,TN,A,CPSKO,SLOPE,CONST,R,WHAT,
     1BEGIN,R1,R2,S)
      Y1=Y1IN/S
      Y2=Y2IN/S
      R1=SQRT(X1**2+Y1**2)
      R2=SQRT(X2**2+Y2**2)
      Z1=R1+X1
      Z2=R2+X2
      IF(R2-R1) 500,501,502
  500 IF(Z2-Z1) 504,504,503
  502 IF(Z2-Z1) 503,504,504
  503 A=ALOG(R1/R2)/(2.*(Z2-Z1))
      CALL BESSEL(A*R1,BIO,BII,BKO,BKI)
      CPSKO=EXP(A*X1)*TN/BKO
      WHAT=1.
      GO TO 505
  501 R=R1
      WHAT=2.
      GO TO 505
  504 WHAT=3.
  505 IF(X1+X2) 506,507,507
  506 BEGIN=-1.
      GO TO 508
  507 BEGIN=1.
  508 SLOPE=(Y2-Y1)*S/(X2-X1)
      CONST=Y1IN-SLOPE*X1
      RETURN
      END
      SUBROUTINE ROWPS(X,YIN,A,C,XINTL,TN,S)
C DETERMINATION OF X FROM TN=C*EXP(-A*X)*KO(A*R)
      TABEF(XDIS,BFUTO)=C*EXP(-A*XDIS)*BFUTO-TN
      DRIVE(XDIS,RDIS,BFUTO,BFUTA)=-C*A*EXP(-A*XDIS)*(BFUTO+XDIS/RDIS*
     1BFUTA)
      XA=XINTL
      Y=YIN/S
```

```
302   RA=SQRT(XA**2+Y**2)
      CALL BESSEL(A*RA,BIORW,BI1RW,BKORW,BK1RW)
      X=XA-TABEF(XA,BKORW)/DRIVE(XA,RA,BKORW,BK1RW)
      IF(ABS(X-XA)-.001) 300,300,301
301   XA=X
      GO TO 302
300   RETURN
      END
      FUNCTION DRAJH(C,A,X,Y)
      R=SQRT(X**2+Y**2)
      CALL BESSEL(A*R,BIO,BII,BKO,BKI)
      DRAJH=C*EXP(-A*X)*BKO
      RETURN
      END
      SUBROUTINE BESSEL(Z,BIOT,BI1T,BKOT,BK1T)
      DIMENSION BIORD(60),BI1RD(60),BKORD(60),BK1RD(60),RRED(60)
      DIMENSION TRED(50),TKRD(50),TDRD(50)
      COMMON BIORD,BI1RD,BKORD,BK1RD,RRED
      COMMON TRED,TKRD,TDRD
      COMMON TMELT,TAMB,ALMIN,TDIFF,DISTN,CCONV,CPAD,VEL,THICK
      COMMON TKXTD,TDXTD,TKMTD,TDMTD,NOPRP
      VALF(A,B,C,D,E)=A+(B-A)/(C-D)*(E-D)
      IF(Z-.05) 8,9,9
9     IF(Z-3.) 1,2,2
1     Z2=20.*Z
      I=Z2
      BIOT=VALF(BIORD(I),BIORD(I+1),RRED(I+1),RRED(I),Z)
      BI1T=VALF(BI1RD(I),BI1RD(I+1),RRED(I+1),RRED(I),Z)
      BKOT=VALF(BKORD(I),BKORD(I+1),RRED(I+1),RRED(I),Z)
      BK1T=VALF(BK1RD(I),BK1RD(I+1),RRED(I+1),RRED(I),Z)
      GO TO 7
2     IF(Z-170.) 3,3,4
3     Z1=Z
      GO TO 5
4     Z1=170.
5     BIOT=EXP(Z1)/SQRT(6.28318*Z1)*(1.+1./(8.*Z1))
```

```
      BI1T=BIOT*(1.-3./(8.*Z))/(1.+1./(8.*Z))
      BKOT=SQRT(1.57/Z)*EXP(-Z)*(1.-1./(8.*Z))
      BK1T=BKOT*(1.+3./(8.*Z))/(1.-1./(8.*Z))
      GO TO 7
8     BIOT=1.+Z**2/4.
      BI1T=Z/2.*(1.+Z**2/8.)
      BKOT=ALOG(2./Z)-.5772
      BK1T=1./Z
7     RETURN
      END
      FUNCTION AUTSCL(YD,DN)
      J=YD/DN
      ZARB1=J
      IF(YD-ZARB1*DN) 701,700,701
701   ZARB2=J+1
      AUTSCL=YD/ZARB2
      GO TO 702
700   AUTSCL=DN
702   RETURN
      END
      SUBROUTINE ORIGIN(RF,A,BIN,RINTL,TN,S)
      R1=RINTL/S
      B=BIN/S
102   CALL BESSEL(A*R1,RI01,RI11,RK01,RK11)
      CALL BESSEL(A*(B-R1),RI02,RI12,RK02,RK12)
      TARF=TN*EXP(A*B)*RK02-RK01
      DRIV=A*(TN*EXP(A*B)*RK12+RK11)
      RF=R1-TARF/DRIV
      IF(ABS(R1-RF)-.001) 100,100,101
101   R1=RF
      GO TO 102
100   RF=RF*S
      RETURN
      END
      SUBROUTINE ORBLW(RFS,A,XL,XL1,W,RGS,S)
      RF=RGS
```

```
      CRITE1=100.
802   XW=RF-XL1
      RW1=SQRT(XW**2+XL1**2/4.)
805   FUNT=(RW1+RF-XL1)*ALOG(RF/(XL-RF))-2.*RF*ALOG(RW1/(XL-RF))
      FDRV=ALOG(RF/(XL-RF))-2.*RF/RW1
      RW=RW1-FUNT/FDRV
      IF(ABS(RW-RW1)-.001) 803,803,804
804   RW1=RW
      GO TO 805
803   G=ALOG(RF/RW)
      H=XW+RW-2.*RF
      A=.5*G/H
      CALL BESSEL(A*RW,BI0,BI1,BK0,BK1)
      BK0D=-BK1
      BK1D=-BK0-BK1/(A*RW)
      RWD=XW/RW
      GD=(RW-RWD*RF)/(RW*RF)
      HD=RWD-1.
      AD=.5*(GD*H-HD*G)/H**2
      TABE=BK0+XW*BK1/RW
      DRIV=(AD*RW+A*RWD)*(BK0D+XW*BK1D/RW)+(RW-RWD*XW)*BK1/RW**2
      RFS=RF-TABE/DRIV
      CRITE2=(DRIV*BK0+BK1*TABE)/BK0**2
      IF(CRITE1-CRITE2) 800,800,801
801   CRITE1=CRITE2
      RF=RFS
      GO TO 802
800   S=W/SQRT(RW**2-XW**2)
      RETURN
      END
      SUBROUTINE PNTEND(XA,YA,XB,YB,GRDX,GRDY,TN,NW,NEXT,XRTWN,YRTWN,S)
      SLINE(SLA,SLARG,SLR,SLC)=SLA*SLARG+SLR*SLC
      CIRCLE(RCIR,ARCIR)=SQRT(RCIR**2-ARCIR**2)
      NPTA=XA/GRDX
      NPTB=XB/GRDX
      FNPT=NPTB-NPTA
```

```
      NPT=ABS(FNPT)
      IF(NPT-NW) 601,602,602
 602  NEXT=1
      XRTWN=(XA+XB)/2.
      CALL TYPEN(XA,YA,XB,YB,TN,AF,CK,SLOP,CONT,R,WHAT,REG,RA,RB,S)
      IF(WHAT-1.) 635,636,635
 635  IF(WHAT-2.) 637,638,637
 636  CALL COLUMN(YRTWN,XRTWN,AF,CK,(YA+YB)/2.,TN,S)
      IF(YRTWN-((YA+YB)/2.+GRDY)) 639,639,637
 638  YRTWN=CIRCLE(R,XRTWN)*S
      GO TO 639
 637  YRTWN=SLINE(SLOP,XRTWN,1.,CONT)
      GO TO 639
 601  NEXT=0
 639  RETURN
      END
      SUBROUTINE COLUMN(Y,X,A,C,YINTL,TN,S)
C THIS SUBROUTINE DETERMINES Y FROMTN=C*EXP(-Z*X)*KO(A*R)
      Y1=YINTL/S
 402  R1=SQRT(X**2+Y1**2)
      CALL BESSEL(A*R1,BIOCL,BIICL,BKOCL,BKIICL)
      TAODR=R1*(TN*EXP(A*X)/C-BKOCL)/(Y1*A*BKIICL)
      Y=Y1-TAODR
      IF(ABS(Y-Y1)-.001) 400,400,401
 401  Y1=Y
      GO TO 402
 400  Y=Y*S
      RETURN
      END
```

.05	1.0006	.0250	3.1142	19.9097
.10	1.0025	.0501	2.4271	9.8538
.15	1.0056	.0752	2.0300	6.4775
.20	1.0100	.1005	1.7527	4.7760
.25	1.0157	.1260	1.5415	3.7470
.30	1.0226	.1517	1.3725	3.0560
.35	1.0309	.1777	1.2327	2.5591

.40	1.0404	.2040	1.1145	2.1844
.45	1.0513	.2307	1.0129	1.8915
.50	1.0635	.2579	.9244	1.6564
.55	1.0771	.2855	.8466	1.4637
.60	1.0920	.3137	.7775	1.3029
.65	1.0084	.3425	.7159	1.1668
.70	1.1263	.3719	.6605	1.0503
.75	1.1456	.4020	.6106	.9496
.80	1.1665	.4329	.5653	.8618
.85	1.1899	.4646	.5242	.7847
.90	1.2130	.4971	.4867	.7165
.95	1.2387	.5306	.4524	.6560
1.00	1.2661	.5652	.4210	.6019
1.05	1.2952	.6008	.3922	.5534
1.1	1.3262	.6375	.3656	.5098
1.15	1.3590	.6754	.3411	.4703
1.20	1.3937	.7147	.3185	.4346
1.25	1.4305	.7553	.2976	.4021
1.30	1.4693	.7973	.2782	.3725
1.35	1.5102	.8409	.2603	.3455
1.40	1.553	.8861	.2437	.3208
1.45	1.599	.9330	.2282	.2982
1.50	1.647	.9817	.2138	.2774
1.55	1.697	1.0332	.2004	.2583
1.60	1.750	1.0848	.1880	.2406
1.65	1.806	1.1395	.1763	.2244
1.70	1.864	1.196	.1655	.2094
1.75	1.925	1.256	.1554	.1955
1.80	1.990	1.317	.1459	.1826
1.85	2.057	1.381	.1371	.1707
1.90	2.128	1.448	.1288	.1597
1.95	2.202	1.518	.1211	.1494
2.00	2.280	1.591	.1139	.1399
2.05	2.361	1.666	.1071	.1310
2.10	2.446	1.745	.10078	.1227
2.15	2.536	1.828	.09484	.1151

2.20	2.629	1.914	.08927	.1079
2.25	2.727	2.004	.08404	.10122
2.30	2.830	2.98	.07914	.09498
2.35	2.937	2.196	.07454	.08916
2.40	3.049	2.298	.07022	.08372
2.45	3.167	2.405	.06616	.07864
2.50	3.290	2.517	.06235	.07389
2.55	3.419	2.633	.05877	.06945
2.60	3.553	2.755	.0554	.06523
2.65	3.694	2.883	.05223	.06139
2.70	3.842	3.016	.04926	.05774
2.75	3.996	3.155	.04645	.05432
2.80	4.157	3.301	.04382	.05111
2.85	4.326	3.453	.04134	.04811
2.90	4.503	3.613	.03901	.04529
2.95	4.688	3.779	.03681	.04264
3.00	4.881	3.953	.03474	.04016

50

50.	45.5	2.31225
100.	44.5	2.05584
150.	43.5	1.57996
200.	42.5	1.39509
250.	41.8	1.12003
300.	41.0	1.04429
350.	40.0	.99457
400.	39.2	.89311
450.	38.5	.82752
500.	37.6	.76845
550.	36.8	.73780
600.	36.1	.70351
650.	35.2	.65597
700.	34.5	.60482
750.	33.8	.58294
800.	32.8	.56199
850.	32.0	.54489
900.	31.2	.52739

950.	30.5	.50348	
1000.	29.7	.47212	
1050.	29.0	.44432	
1100.	28.2	.41701	
1150.	27.5	.40462	
1200.	26.8	.39177	
1250.	25.9	.35928	
1300.	25.0	.33571	
1350.	24.5	.31919	
1400.	23.8	.29566	
1450.	23.1	.27598	
1500.	22.2	.24465	
1550.	21.8	.22843	
1600.	21.1	.20136	
1650.	20.6	.14922	
1700.	20.0	.14046	
1750.	19.5	.12877	
1800.	19.0	.13384	
1850.	18.8	.15280	
1900.	18.1	.18214	
1950.	17.8	.23510	
2000.	17.2	.22717	
2050.	17.0	.22453	
2100.	16.5	.21793	
2150.	16.1	.21264	
2200.	15.9	.2100	
2250.	15.5	.20533	
2300.	15.2	.20165	
2350.	15.0	.19935	
2400.	14.7	.19855	
2450.	14.5	.19379	
2500.	14.1	.18895	
4.			
15.	190.		
3180.	.0625	28.5	
535.	3180.	1.85	.137E-08

```
2.5       .86       .4
.5        .5        .5        .5
  2
150
150
  1
 20
  6
 15
  1
  .01
 1.5
  3
0.0       0.0
.17       .0988
.675      0.0
.0001
  1
.17       .0288
  2
.1271
.153
.1        .3
```

APPENDIX B.

<u>SERVICE PROGRAM</u>

```
C **** PROGRAM EXPERIMENTAL DATA ****
C
      DIMENSION XCCHRT(50),XCFOR(50),TEXP(50),TND(50),XNFOR(50)
      READ(5,100) TAMB,TMELT
      READ(5,100) ALMIN,SPDCOR
      READ(5,100) DIST,TIME
      READ(5,109) GRX
      VEL=5.*DIST/TIME
      PRINT,VEL
      DISIN=SPDCOR*ALMIN/VEL
      PRINT,DISIN
 1    READ(5,101) NTCPL
      READ(5,101) NRUN
      READ(5,100) XTCPL,XSTP
      READ(5,101) NOP
      READ(5,100) XCHRT,VELR
      READ(5,102) (TEXP(I),I=1,NOP)
      READ(5,109) XCCHRT(1)
      DO 2 I=2,NOP
 2    XCCHRT(I)=XCCHRT(I-1)+XCHRT
      XCONV=300.*VELR/VEL
      XFORC=XCONV*(XSTP+XTCPL)
      PRINT,XCONV
      PRINT,XFORC
      DO 3 I=1,NOP
 3    XCFOR(I)=XFORC-XCCHRT(I)
      DO 4 I=1,NOP
      XNFOR(I)=(XCFOR(I)*GRX)/(XCONV*DISIN)
 4    TND(I)=(TEXP(I)-TAMB)/(TMELT-TAMB)
      WRITE(6,103) NRUN
      WRITE(6,104)NTCPL
      WRITE(6,105)
      WRITE(6,106) (I,TEXP(I),I=1,NOP)
      WRITE(6,107)
      WRITE(6,108) (XNFOR(I),TND(I),I=1,NOP)
      READ(5,101) NGO
```

```
      IF(NGO.GT.0) STOP
      GO TO 1
  100 FORMAT(2F10.5)
  101 FORMAT(I3)
  102 FORMAT(8F10.5)
  103 FORMAT('1EXPERIMENTAL RUN',I3,//)
  104 FORMAT(' THERMOCOUPLE NUMBER',I3)
  105 FORMAT('0EXPERIMENTAL DATA')
  106 FORMAT(' ',I3,3X,F10.5)
  107 FORMAT('0NONDIMENSIONAL DATA',/,' DISTANCE      TEMPERATURE')
  108 FORMAT(' ',F10.5,3X,F10.5)
  109 FORMAT(F10.5)
      END
77.       2720.
.12877     48.
28.5       1.
.5
```

```
PDFT
C
C **** PROGRAM PUDDLE FIT ****
C
      DIMENSION X(55),YB(55),YL(55),COEFB(2),COEFL(2)
      READ(5,100) RHO,ABAR,C
      IR=1
      TAMB=530.
      THICK=.0625
 1    READ(5,110)NRUN,ESTIM
      WRITE(6,106)NRUN,ESTIM
      READ(5,101) VOLT,AMP,SPD,ITEST
      IF(ITEST.GT.0) GO TO 2
      READ(5,104) AM,BM
      HM=RHO*C*(3180.-TAMB)*1728.
      STARN=(3.415*VOLT*AMP*12.)/(6.2832*ABAR*THICK*HM)
      X(IR)=ALOG10(STARN)
      WRITE(6,107) AM,BM
      WRITE(6,102) VOLT,AMP,THICK
      WRITE(6,103) STARN
      BSTAR=(SPD*5.*BM)/(24.*ABAR)
      STARL=(SPD*5.*AM)/(24.*ABAR)
      YB(IR)=ALOG10(BSTAR)
      YL(IR)=ALOG10(STARL)
      WRITE(6,105) BSTAR,STARL
      IR=IR+1
      GO TO 1
 2    IR=IR-1
      CALL LSFIT(IR,2,X,YB,COEFB)
      AB=COEFB(1)
      CNSTB=10.**AB
      WRITE(6,108) CNSTB,COEFB(2)
      CALL LSFIT(IR,2,X,YL,COEFL)
      AL=COEFL(1)
      CNSTL=10.**AL
      WRITE(6,109) CNSTL,COEFL(2)
```

```
100   FORMAT(3F12.7)
101   FORMAT(3F10.5,I2)
102   FORMAT(' VOLTS=',3F10.5)
103   FORMAT(' NSTAR=',F10.5)
104   FORMAT(2F10.5)
105   FORMAT(' BSTAR=',F10.5,' LSTAR=',F10.5)
106   FORMAT('ORUN NUMBER',I2,3X,A4)
107   FORMAT(' LENGTH=',F10.6,' WIDTH=',F10.6)
108   FORMAT('0',///,' BSTAR VS. NSTAR',/,' CONSTANT=',F12.6,' EXPONENT
     1=',F12.6)
109   FORMAT('OLSTAR VS. NSTAR',/,' CONSTANT=',F12.6,' EXPONENT=',F12.6)
110   FORMAT(I2,A4)
      STOP
      END
      SUBROUTINE LSFIT(N,M1,X,Y,B)
      DIMENSION X(55),Y(55),A(100),B(M1),C(55,10)
      IPRNT=6
      DO 1 I=1,N
    1 C(I,1)=1.0
      DO 2 J=2,M1
      DO 2 I=1,N
    2 C(I,J)=C(I,J-1)*X(I)
      DO 3 J=1,M1
      DO 3 I=1,M1
      L=I+(J-1)*M1
      A(L)=0.
      DO 3 K=1,N
    3 A(L)=A(L)+C(K,I)*C(K,J)
      DO 4 I=1,M1
      B(I)=0.
      DO 4 K=1,N
    4 B(I)=B(I)+C(K,I)*Y(K)
      CALL SIMQ(A,B,M1,I)
      IF(I.EQ.0) GO TO 5
      WRITE(IPRNT,100)
 100  FORMAT(1H1,'  SINGULAR COEFFICIENT MATRIX PASSED TO SIMQ -- LSFIT
```

```
     1RESULTS WILL BE UNRELIABLE',//)
   5 RETURN
     END
     SUBROUTINE SIMQ(A,B,N,KS)
     DIMENSION A(100),B(N)
     TOL=0.
     KS=0
     JJ=-N
     DO 65 J=1,N
     JY=J+1
     JJ=JJ+N+1
     BIGA=0.
     IT=JJ-J
     DO 30 I=J,N
     IJ=IT+I
     IF(ABS(BIGA).GE.ABS(A(IJ))) GO TO 30
     BIGA=A(IJ)
     IMAX=I
  30 CONTINUE
     IF(ABS(BIGA).GT.TOL) GO TO 40
     KS=1
     RETURN
  40 I1=J+N*(J-2)
     IT=IMAX-J
     DO 50 K=J,N
     I1=I1+N
     I2=I1+IT
     SAVE=A(I1)
     A(I1)=A(I2)
     A(I2)=SAVE
  50 A(I1)=A(I1)/BIGA
     SAVE=B(IMAX)
     B(IMAX)=B(J)
     B(J)=SAVE/BIGA
     IF(J.EQ.N) GO TO 70
  55 IQS=N*(J-1)
```

```
      DO 65 IX=JY,N
      IXJ=IQS+IX
      IT=J-IX
      DO 60 JX=JY,N
      IXJX=N*(JX-1)+IX
      JJX=IXJX+IT
   60 A(IXJX)=A(IXJX)-A(IXJ)*A(JJX)
   65 B(IX)=B(IX)-B(J)*A(IXJ)
   70 NY=N-1
      IT=N*N
      DO 80 J=1,NY
      IA=IT-J
      IB=N-J
      IC=N
      DO 80 K=1,J
      B(IB)=B(IB)-A(IA)*B(IC)
      IA=IA-N
   80 IC=IC-1
      RETURN
      END
```

.283 .500 .105

APPENDIX C.

RECORD OF EXPERIMENTAL DATA

A. Weld Puddle Correlation Data

Run No.	E	I	V	w	y	w*	y*	N*	EVALUATION
1	15	210	37.8	.602	.210	9.49	3.31	4.83	Poor
2	15	210	37.8	.523	.200	8.24	3.14	4.83	Poor
3	15	210	37.8	.615	.210	9.68	3.30	4.83	Poor
4	15	210	37.8	.525	.186	8.27	2.93	4.83	Fair
5	15	210	37.8	.600	.216	9.38	3.40	4.83	Fair
6	15	210	37.8	.600	.200	9.45	3.15	4.83	Fair
7	18	250	48.7	.682	.207	13.84	4.20	6.90	Good
8	18	240	42.5	.591	.206	Not Computed		- - - -	Very Poor
9	16	260	42.5	.600	.205	10.63	3.64	6.38	Poor
10	15	200	32.4	.584	.229	7.89	3.10	4.60	Poor
11	15	200	32.4	.576	.211	7.77	2.84	4.60	Poor
12	15	205	32.4	.575	.201	7.76	2.72	4.72	Fair
13	15	210	32.4	.575	.184	7.76	2.48	4.83	Poor
14	15	210	32.4	.632	.198	8.53	2.67	4.83	Poor
15	15	190	28.5	.598	.185	7.10	2.20	4.37	Fair
16	15	190	28.5	.583	.192	6.92	2.28	4.37	Tood
17	17	280	48.7	.728	.209	14.77	4.24	7.30	Good
18	17	280	48.7	.736	.205	14.93	4.16	7.30	Fair
19	16.5	235	42.5	.741	.213	13.12	3.78	5.95	Good
20	16.5	235	42.5	.784	.190	13.87	3.37	5.95	Good
21	16	230	40.5	.755	.210	12.61	3.51	5.64	Good
22	16	230	40.5	.728	.183	12.16	3.06	5.64	Good
23	15	210	37.8	.621	.154	9.78	2.43	4.83	Good
24	15	210	37.8	.650	.191	10.24	3.01	4.83	Fair
25	15	210	37.8	.606	.172	9.54	2.71	4.83	Fair
26	15	205	32.4	.684	.180	9.23	2.42	4.72	Good
27	15	205	32.4	.627	.196	8.46	2.65	4.72	Fair
28	15	190	28.5	.684	.198	8.12	2.35	4.37	Good

B. Experimental Temperatures

Run #1

Distance from Local Origin (in.)	Temperature (°F.) at Thermocouple 1	2	3	4	5	6
0.0	77	77	77	77	77	77
0.5	510	330	120	90	92	90
1.0	875	730	240	205	290	170
1.5	1130	1005	610	490	605	355
2.0	1285	1160	810	690	725	605
2.5	1370	1285	960	790	840	710
3.0	1410	1310	1065	890	920	805
3.5	1410	1370	1095	970	970	890
4.0	1380	1355	1175	1010	1005	955
4.5	1355	1335	1190	1050	1030	990
5.0	1330	1305	1190	1070	1045	1018
5.5	1290	1290	1185	1085	1050	1030
6.0	1270	1260	1175	1085	1050	1040
6.5	1240	1240	1165	1070	1040	1050
7.0	1210	1205	1145	1065	1030	1055
7.5	1190	1180	1135	1050	1025	1050
8.0	1170	1150	1125	1040	1020	1040
8.5	1150	1130	1105	1025	1010	1040
9.0	1120	1115	1095	1010	995	1030
9.5	1090	1090	1080	1000	990	1025
Distance from Weld C/L. (in.)	.127	.139	.153	.181	.190	.225
XCCHRT(1)	9.74	12.68	15.78	19.15	22.18	24.95

XCCHRT(1) is the distance in inches from the origin of the run to the local origin of each curve.

APPENDIX D.

ANALYTICAL DETAILS

A. Origin of the Nondimensional Welding Parameter, N*

The Rosenthal solution for the 2-D case assuming a thin plate and a line source is:

$$T - T_o = \frac{\dot{Q}}{t} \frac{1}{2\pi\bar{k}} e^{-\bar{\lambda}Vw} K_o(\bar{\lambda}Vr) \qquad \text{(D-1)}$$

where $\dot{Q} = 3.415\ \eta EI$ (Btu/hr)

and $r = \sqrt{w^2 + y^2}$ (ft.)

H_m, the heat content of the material, is defined:

$$H_m = \overline{c\rho}\,(T_m - T_o)$$

which ignores the heat required for the pearlite-austenite transformation and the latent heat of fusion which are small effects and very difficult to measure accurately. By multiplying and dividing equation (D-1) by $\overline{\rho c}\,(T_m - T_o)$ the following results:

$$T - T_o = \eta \frac{3.415EI}{t} \frac{1}{2\pi} \frac{\overline{\rho c}}{\bar{k}} \frac{T_m - T_o}{\overline{\rho c}(T_m - T_o)} e^{-\bar{\lambda}Vw} K_o(\bar{\lambda}Vr) \qquad \text{(D-2)}$$

which reduces to:

$$T^* = \eta \frac{3.415\ EI}{2\pi t\ H_m\bar{\alpha}} e^{-w^*} K_o(r^*)$$

when the nondimensional quantities defined in sections III and IV are included. If a nondimensional welding parameter is defined:

$$N^* \equiv \frac{3.415\ EI}{2\pi\bar{\alpha}\ t\ H_m}$$

then the Rosenthal solution in the nondimensional form is the following:

$$T^* = \eta\ N^*\ e^{-w^*} K_o(r^*)$$

The same parameter was used for this study since it combines plate

thermal properties, thickness, welding voltage, and welding current in a convenient form.

Thesis 134538
L675 Lipfert
Temperature histories in thin steel plate during gas metal arc welding.